Editors:
Adam Wojciechowski
Piotr Napieralski

COMPUTER GAME INNOVATIONS

2017

Lodz University of Technology Monograph

Wojciechowski, Napieralski (Eds.)

Computer game innovations

2017

Lodz University of Technology, Poland
Faculty of Technical Physics, Information Technology and Applied Mathematics
Institute of Information Technology

Computer Game Innovations

Edited by:
Adam Wojciechowski and Piotr Napieralski

Lodz University of Technology Monograph
Łódź 2017

Reviewed by: Prof. DSc PhD Ewa Napieralska-Juszczak
DSc PhD Przemysław Ignaciuk
DSc PhD Adam Niewiadomski
DSc PhD Maria Pietruszka
DSc PhD Aneta Poniszewska-Marańda
DSc PhD Wojciech Tylman
DSc PhD Danuta Zakrzewska
DSc PhD Mariusz Zubert
PhD Jarosław Andrzejczak
PhD Radosław Bednarski
PhD Agnieszka Duraj
PhD Krzysztof Guzek
PhD Michał Karbowańczyk
PhD Rafał Kotas
PhD Krzysztof Lichy
PhD Przemysław Nowak
PhD Michał Morawski
PhD Joanna Ochelska-Mierzejewska
PhD Marcin Ostrowski
PhD Mariusz Postół
PhD Dariusz Puchała
PhD Jan Rogowski
PhD Wojciech Sankowski
PhD Bartłomiej Stasiak
PhD Kamil Stokfiszewski
PhD Dominik Szajerman
PhD Arkadiusz Tomczyk
PhD Agnieszka Wosiak
Cover: PhD Jarosław Andrzejczak
Technical editor: PhD Dominik Szajerman

LODZ UNIVERSITY OF TECHNOLOGY PRESS
90-924 Łódź, 223 Wólczańska Street
phone: +48 42 631-20-87
fax: +48 42 631-25-38
e-mail: zamowienia@info.p.lodz.pl
www.wydawnictwa.p.lodz.pl

ISBN 978-83-7283-868-1
Edition 70 copies. Offset paper 80g
Printed by: Offset printing “Quick-Druk” s.c. 90-562 Łódź, 11 Łąkowa Street
Nr 2254

Contents

The learning of programming through the creation of games by children aged 7–15

Maria Pietruszka

Faculty of Economics and Sociology at the University of Lodz, Łódź, Poland

Abstract

The current curriculum of IT education in Polish primary and lower secondary school was not adjusted to the level of ability of modern students who have been in contact with computers, tablets and smart-phones from an early age. These devices are used by children for entertainment and information retrieval. This is a chance to interest them in learning programming through game programming. In addition, it coincides with the new core program for the 8-grade elementary school, which introduced algorithms and programming. The article presents current state of IT education in Polish schools, various initiatives and programs to learn programming through game development, and research results from parents and children from three age groups: 7-9 years (The Kodu Game Lab), 10-12 years (The Scratch) 13-15 years (The Unity).

Index Terms

game development, programming in elementary school, new 2017 programmatic foundation, programming initiatives, usability of game software for children

1 Introduction

Computer science in Polish schools was introduced in the 1960s. However, the curriculum and regular IT classes appeared scarcely in the 1980s. The first IT classes involved programming. Over time, they have been replaced by text editing, multimedia presentations, drawing, and online information retrieval. Currently, few schools teach children programming, which is usually limited to the Logo language for younger children and the basics of Pascal or C for the older. The new programming base is obliged to change this [10]. Today's children find IT lessons boring and unwieldy, computers and mobile devices use to search for information on the Internet and games. According to the study #I'm a player, conducted by the Ipsos, it appears that 86% of every day players, are the players under 15th years of age. Children are eager to tap into the game because they stimulate their imagination and give them the opportunity to succeed in the virtual world, which can enhance their self-esteem in the real world. Can it be combine what children like with the curriculum requirements? Already in the early 1950s a new concept of education – edutainment or entertainment education appeared. It is a combination of education with elements of entertainment [6]. The advantage of such an approach is the concealment of the educational message in play, so that the child does not always know that while playing, he develops his knowledge. This concept can be used to learn programming when the child creates his or her own game. The next chapter presents the stages of child cognitive development and the related educational concepts. After that it analyzed sftware and initiatives to encourage children to learn computer science and game development. Finally, there were presented the usability of selected software for children in three different age groups.

2 IT EDUCATION AT SCHOOL

Until the current year the process of IT education in primary school was divided into three phases corresponding to the competencies acquired one after another [7]:

- Computer literacy, the basis of computer use.
- Proficiency in using modern technologies.
- Commutative thinking involves methods and tools that a computer user can formulate and solve with more complex problems than with previous steps.

Much attention was paid primarily to the computer applications skills and the use of online resources and communication. Algorithm and programming elements were implemented to a small extent or not at all, appearing only in junior high school or later (Table 1) [10].

TABLE 1
Comparison of the current IT core (I-VI) and extended (junior high school or high school) [10]

	Basic education	**Extensive education**
Knowledge of algorithms and programming	Algorithms and programming elements implemented to a small extent or not at all	Advanced knowledge of algorithms and programming
Knowledge of usable applications	Skillful knowledge of applications, often the same content and skills at different stages	Troubleshooting supported with knowledge of applications with programming elements
Teacher preparation	Often teach tutors from other subjects with additional powers	Classes are provided by well-trained teachers

The new core curriculum for 8-grade primary school states that “Developing children’s and young people’s IT competencies is one of the priorities of state education policy” [10]. Among the seven most important skills appeared “creative problem solving in various fields with the conscious use of computer science methods and tools, including programming” [10]:

- Class I-III – Visual programming of simple situations or stories, sequences controlling the object on the screen,

- Classes IV-VI – Basics of algorithms and visual programming using sequential, conditional and iterative commands and events,
- Class VII-VIII – development of algorithms, textual programming with input/output instructions, arithmetic and logic expressions, iterative instructions, functions, variables and arrays.

No programming tools have been imposed. It is the teacher's decision to choose a programming environment appropriate for implementing the core curriculum guidelines and taking into account the mental development of children.

Jean Piaget, a Swiss psychologist, in his research put the children in a series of questions that aimed to get to know their thinking processes and reasoning [5]. On this basis, he presented the theory of child cognitive development in which he distinguished four stages:

1) Sensorimotor stage (until the age of 2) [4].
2) Preoperational stage (2-7 years) – the child stops thinking only in a sensorimotor manner, but only after the age of six begins to understand that the perspective of others may differ from his or her perspective and learns of focus on more than one aspect of the event/stimulus [8].
3) Concrete operational stage (7-11 years) – mental processes gradually become logical, the child is able to solve problems on the basis of reasoning, but not necessarily word-of-concept demands for future.
4) Formal operational stage (11-15 years) – the ability to logical thinking, reasoning and solving various problems. Thinking of a child is similar to that of an adult [8].

Therefore, it is necessary to differentiate the programming method. Young children from the first three classes have problems with logical and abstract thinking and formulating their own thoughts into sentences. They think in terms of obvious, concrete. Therefore, games designed for them should not contain complex commands and texts, but offer visual programming with picture puzzles. Children in grades IV-VI begin to draw conclusions and think abstractly. Development environments for such children should already be able to build scripts by selecting blocks of code from the list and dragging them into the script area. Children aged 13-15 think like adults, so it is the age appropriate for typing code.

3 Software and Initiatives to Encourage Children to Learn Programming

When selecting software for children, the following criteria should be considered:

- Availability – programs should be made available to children so that they can continue their education at home or at school on IT interest group. From the point of view of school and parents, it is important that the programs can be downloaded free of charge.
- User Community – provides access to software documentation and training materials, sharing ready-made games with other users.
- Difficulty level – should be appropriate for different age groups. As a rule, developers of software or teaching materials give what age they are intended

for. Friendly – too complex interface and a large range of program capabilities can discourage the child.

The combination of games and programming pulls in and makes learning a pleasure. Therefore, most of the programs presented below are based on the above concept.

For the youngest children, the LightBot and the Code Game Lab software are suitable. LightBot is dedicated to even 4-year-old. Using the icons by a child, the robot is programmed to fire all the blue tiles on each level. Typing of coding is not required. Moving the robot forward, rotating, jumping and lighting the tiles is programmed by selecting the appropriate icon in the main procedure (Fig. 1). The child learns to use sequences, conditions, and loops.

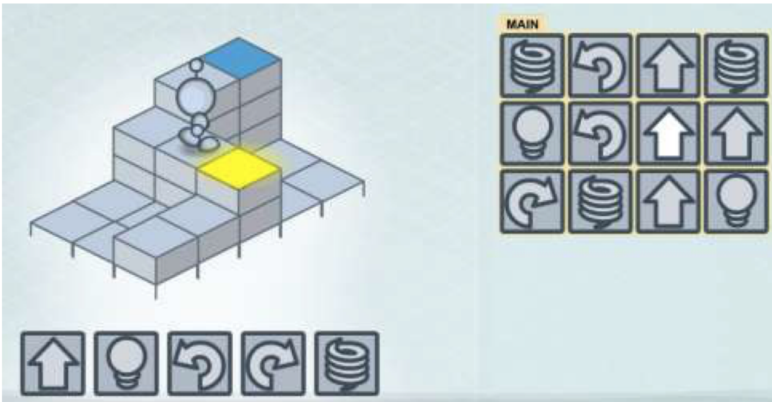


Fig. 1. LightBot; https://lightbot.com

Microsoft Kodu Game Lab is an environment for designing, creating, and testing your own 3D games for PCs and Xbox. Using the tools menu, you create a world filled with mountains, lakes and rivers. Then objects (e.g. trees, huts, rocks and figures) are added. In order for a character to be able to move around and react to events, he uses the Kodu puzzles – a visual programming language. Puzzles are placed in two blocks. The “When” block is used to define an event, such as pressing an arrow key on a keyboard or a collision of a programmed character with another object. The “Done” block determines what character have to do if an event occurs (Fig. 2).

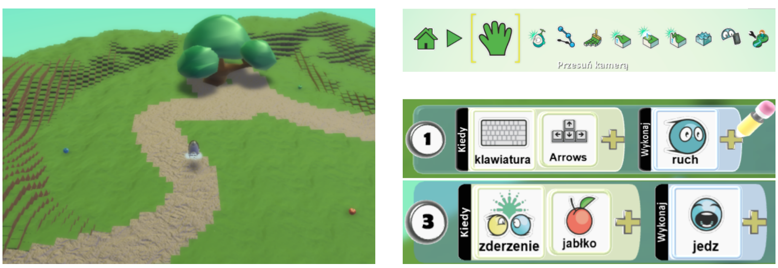


Fig. 2. MS Kodu Game Lab: game world (left), tools menus (top-right), blocks of event base programming (bottom-right); https://www.kodugamelab.com/

Visual programming languages for older children (10-12 years) are more advanced. Code blocks in natural language are drag & drop to the script field. On that purpose is created the game by Scratch – a visual programming language designed by Mitchel Resnick at MIT – Massachuset Institute of Technology Media Labs (USA). In Scratch, the characters are called sprites, the background of the game – the scene, and the code created – the script (Fig. 3). The user has various categories of code blocks available: motion and appearance of sprites, sound and pen, events and sensors, data and expressions. The most important blocks are in control category because they are used to add loops and condition statements to scripts and to stop scripting. Blocks have different shapes and colors, so the structure of the script is legible.

Fig. 3. Scratch: scene with sprites (left), script code for moving the sprite with the arrow keys (right); https://scratch.mit.edu/

The most well-known programming initiative is Code.org's Hour of Code, which has so far attracted the attention of 200 million children from over 180 countries. New hourly tutorials are available in 45 languages, also in Polish. Children learn how to program games with characters from their favorite games and movies: the Minecraft, the Star Wars, the Frozen, and the Angry Birds (Fig. 4). Just like in Scratch, programming involves choosing the right block of code from the list and dragging it into the script area. The organization's website also includes 20-hour tutorials for children from 4 to 18 years old.

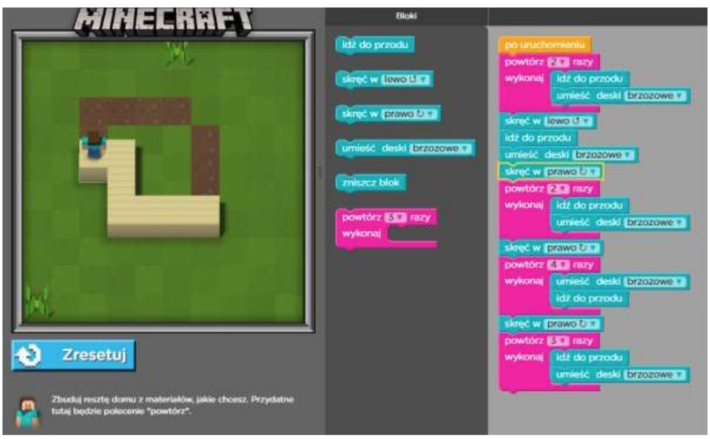


Fig. 4. The Hour of Code with Minecraft; https://code.org

Many development environments for children and adolescents use turtle geometry. The turtle is a robot that reacts to a few simple commands, such as "LEFT 45", which means turning left by 45 degrees [1]. The tortoise movement leaves a trace on the screen in the form of geometric drawings. In the 1960s, Seymour Paperton, director of the Logo Group at the Massachusetts Institute of Technology (MIT), joined turtle graphics into Logo programming language. Logomotion is a Polish multimedia version of the Imagine Logo, developed in 2002 at the Center for Computer Education and Computer Applications (OEIiZK), recommended since 2003 as a means of education by the Minister of National Education and Sport. By controlling the tortoise, pupils of classes I-III develop the ability to define direction and shape mapping, perception and independent thinking (Fig. 5). Older children can benefit from advanced program options – each turtle can have its own variables, procedures and event handlers. The demo version is free.

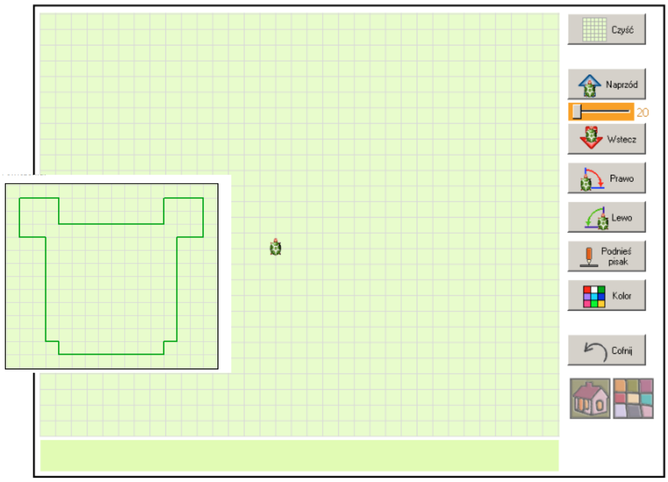
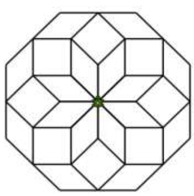


Fig. 5. Logomocja: the application of the Sterujemy.imp to model the pattern with turtle movements [2]

Turtle graphics are also available in Microsoft Small Basic (Fig. 6). This is a very simple programming environment for creating applications in a text box and graphical window. In addition to learning basic programming mechanisms, this introduces syntax and basic instructions in Basic language. The program has a built-in IntelliSense function that prompts snippets of code, explains their meaning and syntax. It is designed for the children aged 10-16 and beginners learning programming.



```
For j=1 To 8
  For i=1 To 8
    Turtle.Move(50)
    Turtle.Turn(45)
  EndFor
Turtle.Turn(45)
EndFor
```

Fig. 6. Microsoft Small Basic: turtle graphics and its code; http://smallbasic.com/

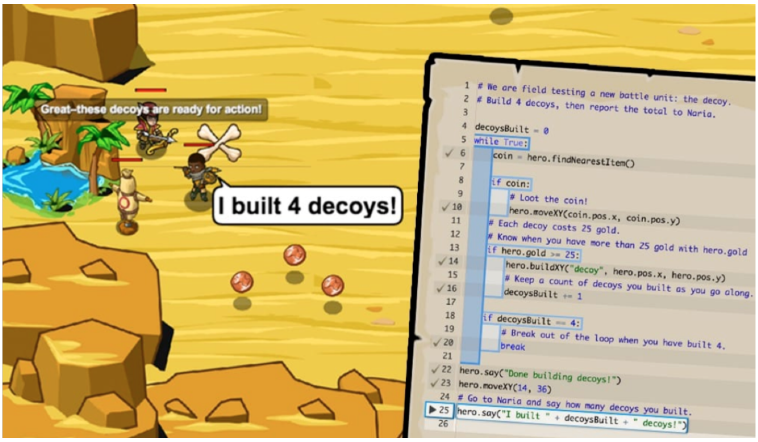


Fig. 7. Code Combat; https://codecombat.com/

Code Combat is a platform for students to learn programming by playing an interesting RPG game (Role-plaing game). At the beginning, the player can choose from several available characters to play. It can also choose the language in which the character's movements will be programmed. When typing code, the program instructs the player what instructions he can use and indicates errors. The effect of the code is immediately visible on the screen (Fig. 7). The user who completes the level receives rewards in the form of gems, items and points. The following programming languages are available:

- Python and JavaScript.
- Lua – scripting language with syntax similar to Pascal.
- CoffeeScript – a simple language compiled into JavaScript with syntax similar to Python and Ruby.

When performing individual tasks, the player learns the syntax of the language, loop statements, decision instructions, variables, and other programming elements. Teacher assistance materials are available on the website.

Unlike previous software, Unity is not a programming-oriented initiative but a popular computer game engine. However, Personal Edition is free. For publishing a commercial game you pay when your sales revenue exceeds $ 100,000. Games are supported on 21 platforms, including Windows, Mac, Linux, Android, iOS, PlayStation, Xbox, Oculus Rift goggles. You can use the toolbar to manipulate objects (Fig. 8 down). The basic concepts are GameObject – each object added to the game (e.g. player character, opponents, weapons), Component – component of the object, Assets – resources, Scene – area of the game, e.g. level or menu, Prefab – a prefabric, that is, a matrix to create many the same objects, Script – code in C# language. Character control requires creating a new script and attaching it to character properties.

The concept of a computer-free lesson appears in the new IT education curriculum. It is worthwhile to reach for such initiatives as the Computer Science Unpluggd. It is a collection of computer science classes for children of all ages. Through logical games, games and puzzles, they teach children such issues as converting binary to decimal numbers, algorithms, artificial intelligence, computer graphics basics, machine run. Didactic materials are available under Creative Commons license.

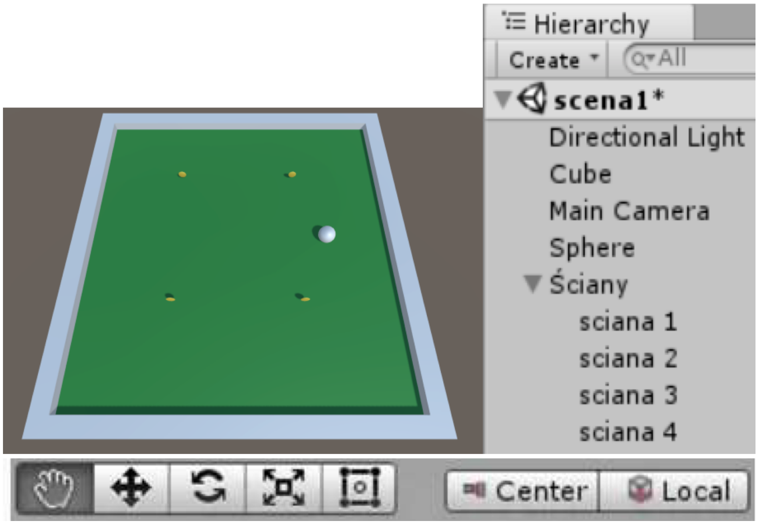


Fig. 8. Unity: scene with objects (left), object hierarchy (right), control bar with tools for moving around the scene, and for translation, rotation, scaling and transformation of free objects (down)

4 Usability Testing with Children

According to ISO 9241-11 usability is "extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (in http://www.userfocus.co.uk/resources/iso9241/part11.html). The purpose of software usability testing is to verify that users are able to use the program themselves and observe their behavior and interaction. Usability studies in children group (especially aged 7-9 years) differ from studies conducted in adults group [3]. Children may not understand the question. The person conducting the study could formulate them in a way that was too complicated for the child, using incomprehensible words or the question could be too long. Children often have a problem with the correct expression of their thoughts, because their vocabulary may be insufficient. Therefore, the evaluation of the program under review may not be adequate to what child really thinks about and therefore unreliable. It is important to consider the behavior of children during the task. If the child is reluctant to execute the command, it drills, turns away from the computer, it may mean that he is not satisfied with working with the program or just bored. The younger the children, the harder is to focus their attention on their task. In addition, it is a good idea to avoid situations in which a child feel like at school because it will try to do the job or answer questions in the way the examiner expects.

The following programs were selected for the study: Kodu Game Lab for children aged 7-9, Scratch for 10-12 years, Unity for 13-15 years. The first two met all the criteria listed in the previous chapter. Scratch is also used in some Polish schools. The

Unity program was chosen at the request of the oldest children in spite of the high level of difficulty and too many opportunities offered.

The study was divided into three stages:

- Conduct lessons according to the outline developed to familiarize children with all stages of game creation.
- Make your own.
- Completion of the survey in which children evaluated the program, responded to what they liked/disliked, which was easy/difficult.

The game created by the students consisted of collecting objects by the hero (e.g. apples in Kodu Game Lab, mouse in Scratch, coins in Unity). To win in the Kodu Game Lab and Scratch should have collected a certain number of points. The scenarios also included presetting conditions for losing. In Unity, the game was won after collecting all the objects.

All the children studied were willingly performing their tasks during the lesson. Mostly they worked on their own, and in case of problems they asked for help from the teacher.

Sometimes children from the youngest group do not remember what tool to use to do any particular exercise in the Kodu Game Lab. The hardest thing was to program the objects according to their own ideas.

67% of middle school children have already used Scratch. These children did not have troubles programming objects according to their own ideas and use of messages. This has happened to children who learned the program only during the study. Still, all children in both groups have managed to do their own programming tasks and create their own games according to the given scenario.

Unity is a powerful engine with many possibilities. The story of the game was not imposed, but it was enforced: turning the camera, adding objects to the scene, creating folders, adding prefabricated items, materials or scripts, and programming scripts. The last task proved to be the hardest because only one of the eight students was in contact with programming. That is why students receive ready-made scripts for character movement, collecting items and counting points. Therefore, due to time constraints, not all pupils were able to finish the game.

After completing the lesson and performing their games children received surveys.

Children aged 7-9 years in 81% rated Kodu Game Lab very well or rather well. Only one child rated the program rather badly and said that he did not want to learn programming and computer graphics by creating games (Fig. 9). Mostly, they liked the opportunity to create their own game and the simplicity of their creation. They did not like the low number of available objects, the low possibility of creating games and too simple graphics.

Children aged 10-12 also rated 81% Scratch very well or rather well. One child evaluated the program rather badly (Fig. 10). 67% (14 out of 21) children would like to use this program at home. Just like before, they liked the opportunity to create their own game and the simplicity of their creation. This time, they also liked the different possibilities of creating a game. They did not like the difficulty of programming characters and scenes and little objects available in the library. All students declared their willingness to learn programming by creating games.

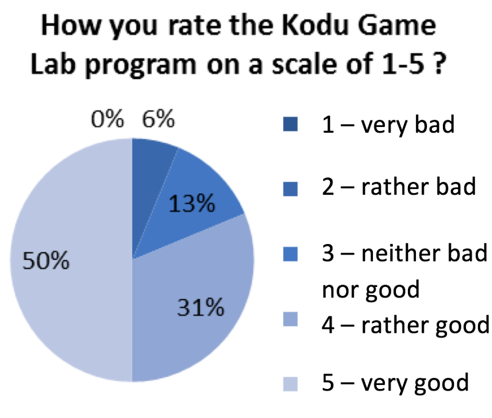


Fig. 9. Assessment of children aged 7-9 years [9]

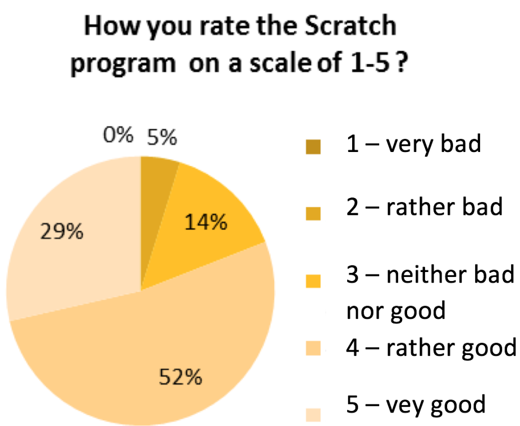


Fig. 10. Assessment of children aged 10-12 years [9]

Compared to previous programs, the creation of Unity games was far more difficult for students. This does not mean that they do not like the program and they do not want to use it. 75% of 13-15 year old rated Unity very well or rather well, 88% would like to use it at home (Fig. 11). Everyone liked the opportunity to create their own game, most of the graphics offered by the program. They did not like the difficulty of programming objects. Still, everyone would like to learn programming by creating games.

Surveys have also been conducted among parents. According to their responses, as many as one third of children spend more than four hours a day at the computer, and almost half of three to four hours (Fig. 12). Contrary to the popular belief, most parents (87%) know what their children play and can name the games. Parents positively value games that develop logical thinking and the ability to solve problems independently. According to them, the main advantage of games is shaping the child's imagination. Only four people have claimed that computer games are time consumers. Most parents would like their children to learn computer programming, computer game development and MS Office programs at computer science lessons at school.

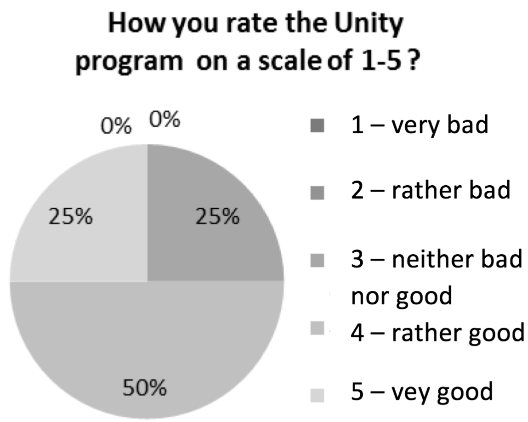


Fig. 11. Assessment of children aged 13-15 years [9]

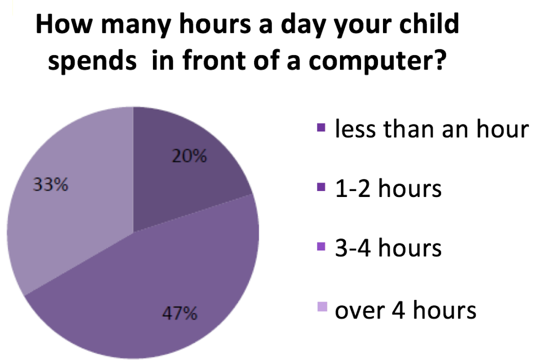


Fig. 12. Parents' answers [9]

5 Conclusions

Program to teach programming should be adjusted to the age of the student. Programs such as Game Lab Code are suitable for learning programming primarily for the youngest children, and for those who have not had any contact with any game development program before. According to older children, this program is too easy to create advanced games and it is quickly boring. The Scratch program, which is currently used in some schools, is better for them. The Unity program has outgrown the capabilities of the oldest group, though it did not discourage kids from learning the game programming. I think in their case, it would be better for them to start the adventure of programming games from Code Combat. A significant group of children declared that they would like to use home-made software to create their own games. Most of the answers to this question were not due to the reluctance to self-develop games but from the lack of a computer at home or consent to use a parent's computer.

Both parents who have expressed a bad opinion about computer games and those who have noticed their good points have generally stated that they would agree to participate their children in programming or computer graphics by creating games.

The sooner your child starts learning the language of programming and troubleshooting with the computer, the better he or she will be in the future. This concerns both those who will continue their adventure with computer science and all others in the future.

It is important to note that even in non-classical coding programs for the youngest (such as the Game Lab Code) the child learns basic programming mechanisms such as conditional statements, loops, events, work on objects.

REFERENCES

[1] Abelson, H.; diSessa, A.A.: *Turtle Geometry. The Computer as a Medium for Exploring Mathematics*. WNT Warsaw 1992 [in polish].

[2] Basaj, H. et al.: *Utilization of modern technologies in early school education*. Center for Computer Education and Computer Applications, Warsaw 2009 [in polish].

[3] Hanna, L.; Risden, K.; Alexander, K.: *Guidelines for usability testing with children*. Magazine Interactions, pp. 9-14, Vol. 4, Issue 5, Sep./Oct. 1997, ACM New York.

[4] Piaget, J.: *The Origins of intelligence in children*. PWN, Warszawa 1966 [in polish].

[5] Piaget, J.; Inhelder, B.: *The Psychology of the Child*. Wyd. Siedmiogród, Wrocław 1993 [in polish].

[6] Singhal, A.; Rogers; E.M.: *Entertainment-Education*. A communication strategy for social change, New York, 1999.

[7] Sysło, M.; Jochemczyk, W.: *Education, computer software in the new basis*. 2009.

[8] Wadsworth, B.J.: *Piaget's Theory of Cognitive and Affective Development*. WSiP, Warsaw 1998 [in polish].

[9] Walak, J.: *Studies on selected games-making software in the group of children aged 7-15*. University of Lodz, Faculty of Economics and Sociology (master thesis conducted by M. Pietruszka) [in polish].

[10] *Program of general education with commentary. Primary school. Computer Science. Center for the Development of Education*. https://www.ore.edu.pl/nowa-podstawa-programowa/, (20 Aug. 2017).

Maria Pietruszka is an associate professor of computer science at University of Lodz and Lodz University of Technology, Poland. Her current research interests are in computer graphics, multimedia, the Web and computer science in economics.

Gesture recognition in virtual reality environments

Roman Chomik, Jarosław Andrzejczak

Lodz University of Technology, Łódź, Poland

Abstract

In recent years there has been great progress in virtual reality systems utilizing 6D motion controllers, but there has been relatively little progress in user interfaces taking advantage of this technology. We present a motion gesture recognizer based on an artificial neural network. By utilizing both convolutional and recurrent layers we have developed a system which needs relatively few training samples to label a wide variety of classes. With reduced input feature count our solution provides comparable success rate to traditional hidden Markov model recognizer. This allows to use our system with motion controllers based on internal measurement units.

Index Terms

gesture recognition, recurrent neural networks, virtual reality

1 Introduction

Gestures are a form of non-verbal communication in which intent is expressed via motion of hands, face or other body parts. Modern virtual reality systems often provide tracking of hands and head movement. This data can be used to infer person's intent. Different devices provide varying precision and amount of data. Systems based on internal measurement units (IMU) provide only acceleration and angular velocity, while more advanced controllers, such as HTC Vive and Oculus Rift, provide absolute positional and rotational information.

A gesture is described by the motion of a body part. Classifying a gesture requires comparing it to previously seen samples. For the purposes of gesture recognition we define a gesture as a series of data points. In general the speed at which a gesture is made does not affect it's meaning. This means that gestures of the same class can vary in length. The features contained in a single data point heavily depend on the specifics of an input device.

In this paper we propose the motion gesture recognizer based on an artificial neural network suitable for rapid development of gesture based user interfaces utilizing modern motion controllers as well as legacy hardware. The three contributions to gesture recognition in virtual reality environments research presented in this paper are:

- Artificial neural network with high recognition rates with varying number of gesture classes and input data format.
- Recognizer that provides high success rate with reduced input feature count compared to traditional hidden Markov model [4].

- Original database for testing gesture recognition in virtual reality containing over two thousand elements (30 gestures repeated 12 times each by 6 participants). Besides twenty gestures identical to those used in [2] ten new, more complex gestures were added.

We start with a discussion of research motivation and related work in the next section. This is followed by a description of a neural network architecture and input data format. We then present the experimental setup details and tests results. Finally, analysis of results and final conclusions will be given.

2 RELATED WORK

There has been some research into motion gesture recognition. Solutions developed so far are based on linear classifiers, AdaBoost and hidden Markov model [2]–[4], [10]. Recognition rates vary depending on type of classifier, number of training samples and testing scenario (user-dependent, user-independent[1]), but in general are claimed to be above 90%.

Linear classifiers work on a principle of manually specifying features based on the interpretation of input data. They are generally based on research by Rubine [1]. Then those feature vectors are compared to reference samples using a linear classifier. This method requires some manual work to define features which give good results. Using 5 gesture samples linear classifier provides recognition rates of up to 96.3% and 95.6% in user-dependent and user-independent scenarios respectively [10].

Hidden Markov model based classifiers improve upon linear classifiers. They do not require prior data processing nor manual specification of features. Without data normalization results vary greatly. In order to give best result data should be normalized prior to feeding it into the classifier which requires the interpretation and analysis of data obtained from input device. Hidden Markov model based classifiers provide good recognition rates of up to 99.8% and 97.0% in user-dependent and user-independent scenarios respectively [4].

3 NEURAL NETWORK ARCHITECTURE

Our recognizer is based on an artificial neural network. Neural networks have been shown to succeed in task where traditional methods have performed poorly or required large workloads, such as speech recognition, language modelling or image analysis. Their main advantage is the ability to learn features, which best describe the nature of the problem. In case of gesture recognition it allows for specifying gestures by example rather than manually specifying features. This in turn makes it possible for the users themselves to add new gestures to the system without exact knowledge of how the system works.

Our neural network has been designed to work well with a variety of input data formats in its raw form. The data is assumed not to be normalized nor in any other way processed. One of the neural network's tasks is extraction of meaningful features from various sensor reading and data representations. It should be able to handle

[1] In user-dependent scenario both training and testing is performed using gesture samples obtained from the same tester. In user-independent scenario training and testing data sets are created using samples from different users or disjoint user groups.

both inertial and absolute readings, and provide acceptable recognition rates with limited input data. To fulfil this task the neural network consist of both convolutional and recurrent layers.

The neural network architecture is layered without skipping, i.e. each layer connects only to the next. It consists of a convolutional layer followed by a number of recurrent layers and a dense projection layer. Each recurrent layer consists of Long Short-Term Memory (LSTM) cells [8]. The projection layer accepts the last output of the recurrent layer and reduces the number of outputs to the number of classes of gestures. Its outputs can be directly interpreted as probabilities of class occurrence in input data. The neural network accepts inputs of varying length. The neural network consists of the following operations in order:

1) Convolution
2) Bias
3) Rectifier activation (ReLU)
4) Dropout
5) LSTM recurrent
6) Dropout
7) LSTM recurrent
8) Dense
9) Softmax activation

Operations 1-3 are referred to as "Convolutional layer". Operations 4-7 are referred to as "Recurrent layer". Operations 8-9 are referred to as "Projection layer" (Fig. 1).

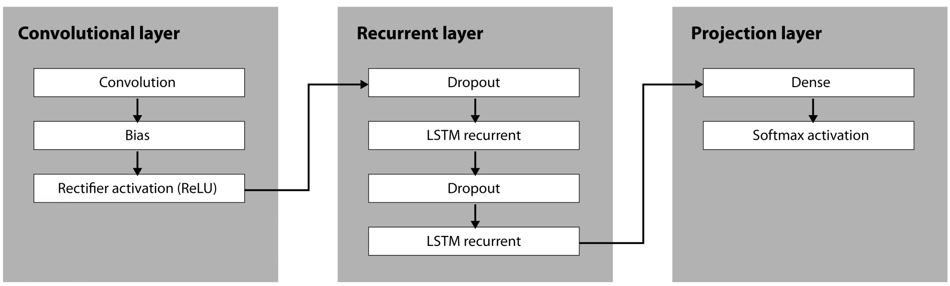


Fig. 1. The neural network operations order scheme showing Convolutional, Recurrent and Projection layer

3.1 Feeding data

The input to our recognizer is represented by a series of vectors. To speed up training multiple gestures are fed in batches. Each vector contains different components recorded at a specific time-point, such as position vector, rotation matrix, etc. Different types are represented as vectors and appended. Vectors in the series are sampled at a constant frequency. As gestures can be of different lengths they are padded with zeros to the length of the longest example in set. This potentially wastes a significant portion of memory, but simplifies training by allowing arbitrary choice of examples in batches. Data in this form is fed directly into the neural network without any further preprocessing.

3.2 Convolutional layer

Convolution has been successfully used in image classification problems [6], [7]. Their structure allows for extraction of local features that can occur in any position in input image. Similarly we have used a convolutional layer to learn local temporal features, i.e. features that occur within short time span. Examples of such features include, but are not limited to, specific change in velocity, position or orientation. Convolution can also potentially be used to detect correlations between different components of input data. Image data is usually two-dimensional (width and height) in nature with a number of color channels (usually 3). Our first dimension is time. In order to allow interactions between different components of input vector we treat components as channels rather than the second dimension. Thus our problem is one-dimensional (time) with number of channels equal to the number of input components.

The first layer accepts the input data. It is treated as an array and convolution is performed. The convolution filter has a width of 4 and a stride of 1 is used. No data padding on edges is applied as it is difficult to perform any meaningful padding without exactly knowing data format and its interpretation, which is exactly the kind of dependency we are trying to avoid. The output of convolutional layer is shorter in length than its input, but contains many more channels. These channels represent learned short term temporal features at given time points in input stream. The next operations is adding a learned bias to each of the features. Then the ReLU activation function is applied. The convolution layer outputs 128 channels.

3.3 Recurrent layer

Because gestures can be of different length and can be executed with different speeds, we need a method that can adapt to varying occurrence of features in input data. Recurrent neural networks have been used to recognize temporal patterns and as such have found use in speech and handwriting recognition, automated translation and language modeling. The recurrent neural network is created by connecting a layer's output as its input in the next time step. The recurrent structure provides them with a limited form of memory. The neural network has a direct access to its current input and indirect access to features that have been available in previous time steps. Because of very large depth of such networks training them via gradient descent methods has been infeasible due to the problem of vanishing/exploding gradient. One solution to this problem is Long Short-Term Memory cells [8]. This works by either permitting entire signal to pass through to the next time step or to be entirely blocked. This results in gradients being passed exactly as is or blocked.

A gesture can be thought of as a sequence of basic movements performed by a user, such as moving the hand in specific direction or twisting the wrist. In each gesture such sequence can be found which is common to all users regardless of the exact way a gesture is performed, the speed of execution or hand wielding the controller. Convolution extracts those basic movements whereas recurrent layer learns how those features contribute into higher level features. By stacking several recurrent layers the neural network is capable of learning entire gestures.

The reccurent layer consists of a two sets of LSTM cells. Each of this sets has a dropout applied to its input to prevent overfitting and improve generalization. The dropout mask is preserved across all timesteps as described in [9]. Dropout keep probability is shared between both sets and has the value of 25% during training and

100% when used for inference. The number of LTSM cell is equal to 128 for each of the sets.

The projection layer is directly used for gesture classification. Based on the last valid output the recurrent layer, the projection layer outputs confidence levels of gesture belonging to each class. The projection layer is comprised of a number of neurons equal to the number of output classes, thus varies depending of data set used. Finally the softmax function is applied. Due to the use of softmax function the sum of outputs is always equal to 1 and individual outputs can be interpreted as probabilities.

3.4 Training

Supervised learning is used to train the model. Training is performed using gradient descent method. In order to provide faster convergence Adam optimizer is used with a learning rate of 0.001 [5]. Training is performed until a predefined training loss is achieved or model no longer converges for a specified number of epochs. No separate validation data set is used. Values of 0.05 and 20 respectively are used for target training loss and number of epochs without convergence. Training is performed on GPU. The size of batch is limited by available memory, thus depends on maximum gesture length and model complexity. In our case batch has a size of 200 gestures. Every epoch the training set is shuffled and new batches are selected. This prevents overfitting to a part of training data and improves generalization.

4 EXPERIMENTAL SETUP

The aim of the performed tests was to examine proposed artificial neural network gesture recognition success rate and identify the minimal number of features needed to obtain high success rate results (90% and more). Additionally we study the effect of number of gesture classes anf their complexity. The neural network model has been implemented[2] using TensorFlow library [11].

Two databases of gesture samples were available for tests. The first database (named Wii) has been created by the authors of [2] and is publicly available. It contains samples generated by 28 testers aged 15-33. Each gesture was performed 10 times by each participant and belongs to one of 20 classes. The second database (named Vive) was created by our team with the assistance of 6 testers using HTC Vive motion controllers. Each participant was asked to perform 30 gestures 12 times each. The first 20 gestures classes are identical to the ones in [2], the remaining are more complex shapes shown on Fig. 2. Due to different data formats, model trained on one database cannot be tested against the other and vice versa.

We have created two experiment scenarios designed to evaluate different aspects of the neural network. The first experiment is designed to directly compare our proposed solution to existing HMM model. The second experiment studies the performance of neural network gesture recognizer with more complex shapes.

The first experimental setup is based on the one described in [2]–[4]. This allows direct comparison of different gesture recognition solutions. The first scenario is user-dependent recognition. The network is trained on half of samples of each class of gestures of a single tester. Then it is tested against the remaining samples of the same

[2]Source code available at https://github.com/romanchom/GestureRecognitionVR

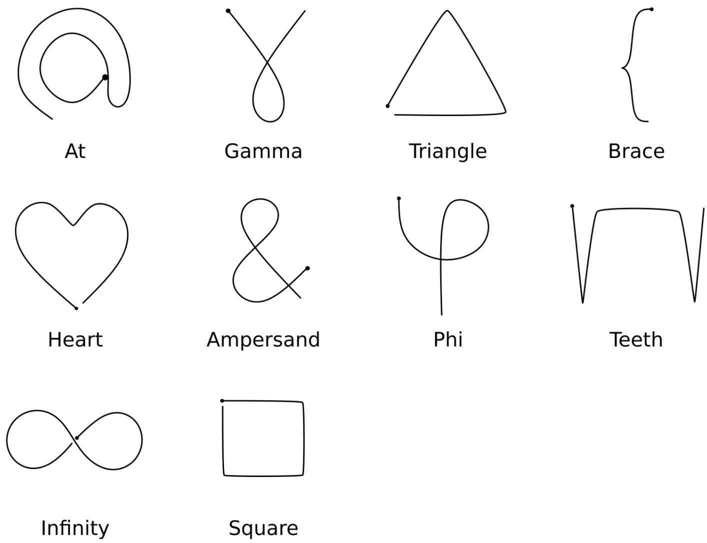


Fig. 2. Shapes of additional gestures in Vive database. Dots denote the beginning of gesture

tester. The experiment is repeated for each of the testers. The second scenario is user-independent recognition. The network is trained on all gesture samples of selected 5 right-handed testers. Then it is tested against all other samples in given database. The experiment is repeated for 100 random combinations of 5 tester groups and the results are averaged. This scenario is designed to show how well the recognizer performs with limited input samples and generalizes to larger data sets. This experiment has been performed for both databases, but only on gestures common to both. The experiment has been performed with each separate feature and with feature sets depending on database used. For the Wii database they exactly the same as in [4] and are as follows:

- PV – position and velocity.
- AW – acceleration and angular velocity.
- AWO – acceleration, angular velocity and orientation.
- PVO – position, velocity and orientation.
- PVOW – position, velocity, orientation and angular velocity.
- PVOWA – position, velocity, orientation, angular velocity and acceleration.

For the Vive database they are as follows:

- PV – absolute position and velocity.
- PVO – absolute position and rotation.
- POVW – position, orientation, velocity and angular velocity.

The second experiment is designed to evaluate the performance of our recognizer with increasing complexity and number of gesture classes. It is performed only on the Vive database in three stages. First only the 20 gestures common to both databases

TABLE 1
User-dependent recognition rates in percent of proposed neural network for single features for both databases and comparison to HMM results without (HMM w/o) and with normalization (HMM n).
P – position, O – orientation, V – velocity, W – angular velocity, A – acceleration

Feature	Wii [%]	Vive [%]	HMM w/o [%]	HMM n [%]
P	98.8	89.3	97.6	97.8
O	98.8	97.0	98.5	98.8
V	98.3	96.1	98.2	98.5
W	98.8	95.8	97.7	98.1
A	98.0	—	98.5	98.4

TABLE 2
User-independent recognition rates in percent of proposed neural network for single features for both databases and comparison to HMM results without (HMM w/o) and with normalization (HMM n).
P – position, O – orientation, V – velocity, W – angular velocity, A – acceleration

Feature	Wii [%]	Vive [%]	HMM w/o [%]	HMM n [%]
P	86.2	81.9	88.7	88.6
O	86.6	80.3	72.6	88.9
V	89.9	93.5	82.0	91.3
W	84.4	91.9	69.8	80.8
A	80.3	—	71.5	88.6

are used. Then only the new gestures are considered. Finally all gestures are used in experiment. We can evaluate how our solution handles more complex gestures and how increasing the number of gestures classes impacts the network's performance. Similarly both user-dependent and user-independent scenarios are considered. Only the full feature set (PVOW – position, velocity, orientation and angular velocity) is used in this experiment.

5 RESULTS

5.1 Comparison to HMM

Table 1 contains the result of the first experiment comparing the proposed neural network to HMM solution when using only single features in user-dependent scenario. Our recognizer performs on par with HMM solution both with and without normalization. The recognition rates are above 98% in the case of Wii database. It is suspected that in the case of user-dependent recognition the intra-class variance is low enough that normalization has very little effect and both systems can perform equally well. The recognition rates obtained using Vive database are somewhat lower and the cause is probably related to the tracking technology.

Table 2 contains analogous results of the user-independent test case. It can be clearly seen that our solution can correctly label a high percentage of samples. The recognition rates are usually higher than those of HMM using not normalized data. Our solution performs on par with HMM using normalized data. In one case it demonstrates that it can perform even better than HMM with normalized data. Despite no knowledge of the underlying data format the neural network is capable of generalizing the gestures with regard to variations introduced by each person's unique way of executing gestures, thus rendering normalization unnecessary.

TABLE 3
User-dependent recognition rates in percent with various feature sets for both databases. PV – position and velocity, AW – acceleration and angular velocity, AWO – acceleration, angular velocity and orientation, PVO – position, velocity and orientation, PVOW – position, velocity, orientation and angular velocity, PVOWA – position, velocity, orientation, angular velocity and acceleration

Feature set	Wii [%]	Vive [%]
PV	98.9	97.4
AW	98.6	—
AWO	98.7	—
PVO	99.4	97.1
PVOW	98.8	97.1
PVOWA	98.8	—

TABLE 4
User-independent recognition rates in percent with various feature sets for both databases. PV – position and velocity, AW – acceleration and angular velocity, AWO – acceleration, angular velocity and orientation, PVO – position, velocity and orientation, PVOW – position, velocity, orientation and angular velocity, PVOWA – position, velocity, orientation, angular velocity and acceleration

Feature set	Wii [%]	Vive [%]
PV	91.9	97.1
AW	88.3	—
AWO	88.9	—
PVO	92.8	94.4
PVOW	90.0	97.7
PVOWA	91.4	—

Tables 3 and 4 contain the result of the first experiment comparing the proposed neural network to HMM system using combined features. In user-dependent scenario the neural network performs well with recognition rates above 98%. It can be seen that the choice of feature set has marginal effect on the success rate and in practice any combination of them could be used. The differences between the two recognizers are within 1% and there is no clear winner between our solution and HMM system using normalized data. In some cases our solution performs better, in others the HMM is superior. In user-independent scenario the results are in favour of the HMM system.

5.2 Effect of gesture complexity and number of distinct classes

The availability of a gesture database containing both simple and complex gestures allowed us to compare the neural network's ability to recognize each. It also lets us study how the number of gesture classes affects the success rate. The results are

TABLE 5
Recognition rates in percent for simple, complex and combined gesture classes in user-dependent (UD) and user-independent (UI) classification

Gesture set	UD [%]	UI [%]
Simple	97.2	92.5
Complex	98.3	97.6
Combined	96.4	94.9

contained in the Table 5. It can be seen that our solution can successfully learn and recognize complex gestures. In fact the recognition rates are higher in case of complex gestures than in case of simple ones. It can be presumed that complex gestures contain more information which increases inter-class variance thus making them easier to correctly classify.

The results using combined gesture classes show that the number of gesture classes has little effect on the recognition rates as long as inter-class variance is sufficiently large. This means that gestures should differ significantly, so that it is difficult for a user to perform an ambiguous gesture.

6 Conclusions

We show that our solution successfully overcomes several challenges. The neural network is capable of learning and extracting features from different feature sets including both absolute and inertial tracking data. With as little as any single feature it is capable of over 98% recognition rate in user-dependent recognition. Moreover data requires no preprocessing and can be feed as is. This means that no additional work is required to decode and normalize data from motion controllers. This makes our recognizer suitable for creating personalized user interfaces based on motion gestures. Any device capable of reading the motion of user's hand can be potentially used as an input device.

In user-independent scenarios our solution performs with around 90% accuracy regardless of device used and feature set available when trained with data from 5 users. Our solution can be applied to different types of motion gestures without modifying the neural network architecture. It can handle both simple and complex gestures at the same time. Our main contribution is the neural network architecture for VR gesture recognition that is extremely flexible. It provides high recognition rates with varying number of gesture classes, input data format and feature count. This makes it suitable for rapid development of gesture based user interfaces utilising modern motion controllers as well as legacy hardware.

References

[1] Rubine, D.: *Specifying gestures by example*. SIGGRAPH Comput. Graph., vol. 25, pp. 329-337, Jul. 1991.

[2] Chen, M.; AlRegib, G.; Juang, B.-H.: *A new 6D motion gesture database and the benchmark results of feature-based statistical recognition*. Emerging Signal Processing Applications (ESPA), 2012 IEEE International Conference on, pp. 131-134, 2012.

[3] Chen, M.; AlRegib, G.; Juang, B.-H.: *6D motion gesture recognition using spatio-temporal features*. Acoustics Speech and Signal Processing (ICASSP) 2012 IEEE International Conference on, pp. 2341-2344, 2012, ISSN 1520-6149.

[4] Chen, M.; AlRegib, G.; Juang, B.-H.: *Feature Processing and Modeling for 6D Motion Gesture Recognition*. Multimedia IEEE Transactions on, vol. 15, pp. 561-571, 2013, ISSN 1520-9210.

[5] Kingma, D.P.; Ba, J.: *Adam: A Method for Stochastic Optimization*. 3rd International Conference for Learning Representations, San Diego, 2015.

[6] LeCunn, Y.; Bottoun, L.; Bengio, Y.; Haffner, P.: *Gradient-based learning applied to document recognition*. Proc. of the IEEE, November 1998.

[7] Krizhevsky, A.; Sutskever, I.; Hinton, G.E.: *ImageNet Classification with Deep Convolutional Neural Networks*. Advances in Neural Information Processing Systems 25 (NIPS 2012).

[8] Hochreiter, S.: *Long Short-Term Memory*. Neural Computation, vol. 9 issue 8, November 15, 1997, pp. 1735-1780.

[9] Gal, Y.; Ghahramani, Z.: *A Theoretically Grounded Application of Dropout in Recurrent Neural Networks*. 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.
[10] Hoffman, M.; Varcholik, P.; LaViola Jr., J.J.: *Breaking the Status Quo: Improving 3D Gesture Recognition with Spatially Convenient Input Devices*. Virtual Reality Conference (VR), 2010 IEEE.
[11] *TensorFlow*. https://www.tensorflow.org/, January 20, 2017.

Roman Chomik Lodz University of Technology department of Technical Physics, Information Technology and Applied Mathematics undergraduate specializing in Game and Computer Simulations Technology. Currently continuing graduate studies at Lodz University of Technology with the speciality of Game Technology and Interactive Systems.

Jarosław Andrzejczak Assistant Professor at the Institute of Information Technology, Faculty of Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology. In 2015 he received PhD degree in computer science for interactive information visualization for digital data sets search results. His research interests encompass: usability testing and engineering, User Experience, data and information visualization, user interface design (including game interface design) as well as application of information visualization in UI design.

Comparison of face animation methods

Radosław Bednarski, Paula Pszczoła

Lodz University of Technology, Łódź, Poland
radoslaw.bednarski@p.lodz.pl, paula.pszczola@gmail.com

Abstract

In this article authors have compared three methods of face animation: clasicc animation with blendshape using, Motion tracking in Blender program and Motion Capture in Autodesk 3ds Max. For each of methods animation of face was created. For all methods performed user tests, which were supposed to indicate the most optimal method for obtaining the most realistic face animation

Index Terms

computer games, face animation, motion capture, motion tracking

1 Introduction

Obtaining an effect where facial animation looks reliably and realistically is a difficult task for any computer graphic. It's about creating animation for the gaming industry, movie industry, and the others. What is the overall process of implementing such animation? It is obtained through geometry manipulation and image manipulation [1]. Geometric deformations mainly concern the shape and facial expressions of a person. Image manipulation helps to reflect the skin and hair properties of the face. Thanks to this, a high level of detail is obtained which is difficult to obtain by manual deformation of the three-dimensional grid. The situation may be different in the case where the three-dimensional model can be so abstract that putting it into motion with the help of a motion capture system can be quite embarrassing. At this point, the animator creates the animation manually. In this paper, the authors compared and tested three methods of facial animation.

2 Related Work

In this chapter, the authors present existring methods of facial animation. Among these will be Blendshape manipulation, Motion Capture and Blendshape, three-dimensional scann and Motion Capture and Motion Capture and realistic mimic features.

2.1 Blendshape manipulation

Face animation requires a deformable three-dimensional model that is able to present a variety of emotions and speech configurations. There are two traditional methods for creating this type of model: a facial model based on facial physiology, and a blendshape model [2]. This first generally simulates the different layers of skin, muscles, adipose tissues, bones, and all components required to approximate the real facial features [3]. The blendshape model, however, generally ignores these properties, but recognizes every facial expression as a linear combination of several facial expressions, ie transition shapes. In order to achieve the full range of facial expressions at the least cost, any weight that can be altered is assigned to each of the combinations mentioned (Fig. 1). This solution guarantees control over the animator model so that getting satisfactory results is facilitated.

Fig. 1. Renders of blendshape models – left: influence on single blendshape model; Middle: the effect of seven blendshape without texture details; Right: influence of seven blendshape with detail texture

2.2 Motion Capture and Blendshape

Below we desribed a method that uses a combination of motion capture and shape variation interpolation. This method includes the flexibility of interpolation and is gaining the fluidity of animation with the help of a motion capture system. This technique consists of several steps: recording facial expressions, decomposing collected data and finally applying the processed data to a three-dimensional model (Fig. 2).

Fig. 2. The top row represents the face of the actress, while the bottom row is the recorded animation for the three-dimensional model [4]

2.3 Three-dimensional scan and Motion Capture

This section will present a method that uses also Motion Capture method in connection with three-dimensional scan [5]. First, an actor's recording is performed using an optical motion capture system that has markers on his face. The analysis of the collected data is then performed to determine the set with the minimum number of facial scans required for the correct facial reconstruction of the three-dimensional model. A two-step process is presented to effectively build a consistent density of surfaces that match all facial scans. They are then reconstructed faithfully reflecting three-dimensional facial movements using a motion capture data connection with a set of minimal facial scans in liquid shape interpolation. The method has been tested on both real and synthetic data. The results show that this method can capture facial movement that fits both face scans and speed captures. The system achieves faithful facial expressions with realistic dynamic wrinkles and detailed details of the actor's face. The main idea of the method is to balance high-precision motion capture and high-resolution face scans to produce a powerful three-dimensional face animation.

Fig. 3. The system captures high-resolution actor moves with dynamically changing realistic wrinkles and other details of the face [5]

2.4 Motion Capture and realistic mimic features

Another method focuses not only on the faithful cast of the actor's face movements on the three-dimensional model, but also on the realistic reflection of the wrinkles on the actor's face during the recording. In the beginning, you need to obtain a static facial scan, including reflections, of the best possible quality. Then recordings are made using an optical motion capture system and 2 synchronized cameras for capturing wrinkles. The final model includes high resolution geometry, motion capture data, and expression wrinkles in a two-dimensional parametric form. This combination is the face shape and its essential elements on many scales. During motion synthesis, motion capture data deforms high-resolution geometry by using linear shear-based mesh deformation. Wrinkle geometry is added to this base model through non-linear energy optimization.

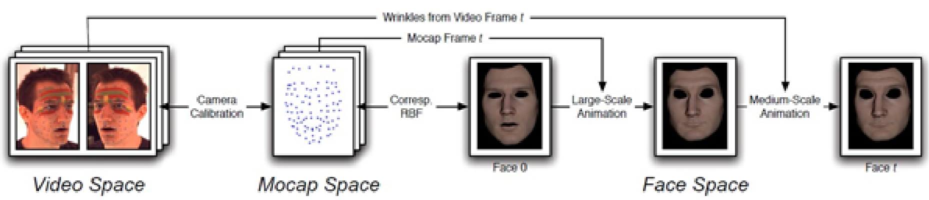


Fig. 4. This method captures video sequences, actor face markers, and a high resolution static face scan [6]

3 Reviving the facial modell

This section shows how to set up a three-dimensional model in motion. In order to accomplish this part, you should used the programs for both modeling and three-dimensional animations: Blender or Autodesk 3ds Max 2016.

Blender was used to created a three-dimensional model from scratch, which was later used for animation (Fig. 5).

To create face animation we used following methods, which will be discussed in the next subsections:

- Classic animation with blendshape.
- Motion tracking in Blender.
- Motion Capture in Autodesk 3ds Max.

Fig. 5. Three-dimensional model using in practical part, author Paula Pszczoła 2017

3.1 Classic animation with blendshape

In this method, a blendshape (Fig. 6) was used to obtain facial animation. The effects of this method depend on the skill of the animator. There are no motion capture utilities or script to control animation. We used only video records references.

Fig. 6. Blendshape created in Blender, Paula Pszczoła Lodz University of Technology 2017

3.2 Motion tracking in Blender

Motion Tracking is a tool in Blender that allows you to capture both two-dimensional and three-dimensional [7] motion in a video recording. It was used as an extension to the previous method, where the animation was created manually. Here there is additional interference in the appearance of motion from the markers movement. This has both its shortcomings and advantages, for example, you can get, for example, quite realistic lip sync, but any imperfections involve manual correction or re-recording. The important thing to pay attention to is what hardware is used to record actor moves. If the camcorder does not have sufficiently high capture frequencies and poor video quality, the recordings may fail because blurred images will be difficult for Blender to accurately track the movement of the markers on the actor's face (Fig. 7).

Fig. 7. From the top view from a regular webcam, down the animation cage obtained with the Motion Tracking tool, Paula Pszczoła, Lodz University of Technology 2017

Advantages:

- The realism of the recorded expressions, except for the exceptions where there was a noise.
- Use of recordings can significantly speed up the entire animation process.

Disadvantages:

- Movement of markers in-depth is not detected.
- The visual effect largely depends on the details of the skeleton.
- Poor recording quality may cause problems with retargeting.
- There is little information on how to properly use Motion Tracking in Blender.

3.3 Motion Capture in Autodesk 3ds Max

In this method we also used blandshape object. Unlike Blender there were another form of their creation. We used a 3ds Max modifier – Morpher which is responsible for transition ftom the base state of the model to the appearance of the blandshape object. This object were created by copying default model and manullay changing its mesh.

In the next steps we used the Analyzer and Face Retargeter – add-on to 3ds max (Fig. 9).

Fig. 8. A set of 28 blendshape objects, used for animations in 3ds Max, Paula Pszczoła, Lodz University of Technology 2017

Advantages:

- No need to apply markers for proper motion capture.
- With an enhanced skeleton, the resulting visual effect is very high.
- Very accurate retargeting with a small amount of animator work.
- Both Analyzer and Retargeter are very easy to use.
- Retargeter, in addition to 3DS Max, is also compatible with Maya, Motion Builder and Softimage programs.
- For the rest of the Faceware software, it is also possible to record and retarget data to a live model (with a plug-in) – FacewareLive [8].

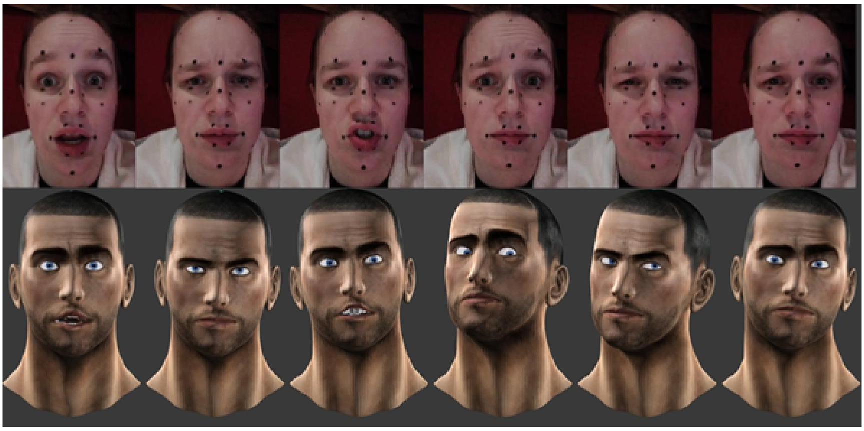


Fig. 9. A set of screenshots showing: upper row, view from a regular webcam; Bottom row, Animation frames obtained through extension to 3ds Max, Faceware Retargeter, Paula Pszczoła, Lodz University of Technology 2017

Disadvantages:

- For the best results, a very accurate GoPro camera and a specially prepared headset for face motion recording may be required.
- The commercial version of the Faceware software requires significant financial backing (local versions of Analyzer and Retargeter cost about 3000 USD) [9].

4 TEST METHOD AND ANALYSIS OF RESULTS

This section will discuss the research that was conducted in the form of a questionnaire and carefully analyzed. This survey was conducted by motion capture specialist, animators and professionals who are interested in animation and know what motion capture is. The test consisted of several questions, both closed and open. It was trying to find the best method for facing animation. The respondents were to choose one of the three methods used to prepare the test material in the form of a video (Fig. 10). This film was in the form of a juxtaposition of the aforementioned methods. Additionally, to make the survey give objective results, it was not told which method was used at which animation (Fig. 11).

The first question in the survey to be answered was which of the animations is the most realistic

In the diagram above, voices are split between two animations, the one on the left dominates. The middle has not received a single voice, because when rendering the animation in Blender, there have been some errors, where through the fragments of the interior of the mouth exhaled through the cheeks of the model.

The second question was a request for a brief justification of the choice of animation. In most cases, the choice was determined by the amount of errors and the fluidity of the transition from one expression to another. Most people, by choosing both left and right animation, noticed the realism of the end result, and in some cases also highlighted what bothered them with the other two animations. Some

Fig. 10. The frame of the movie used in the test, Paula Pszczoła, Lodz University of Technology 2017

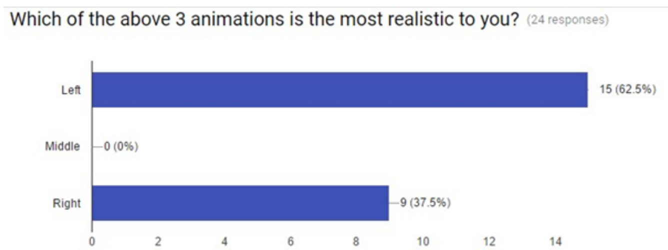


Fig. 11. A graph showing the answers to the first question, Paula Pszczoła, Lodz University of Technology 2017

respondents noted that the central animation is affected by the problems mentioned earlier, and that the right has a rather chaotic eyeball movement. People choosing the animation on the right, mainly focused on the fluidity of movement and lightness with which the model passed from one expression to the other.

The third question referred to the first two and indicates which of the three methods has been used to create the selected animation in the first question. The fourth question requires the justification of the answer to question 3.

According to this diagram the Motion Capture system dominates in the answers.However, after a thorough analysis of the response, it turned out that a large number of people, wrongly attributed the method to the animation. Respondents who asked the animation on the left in the first question, in most cases (12 out of 15), referred to the method of motion capturing, explaining their choice of results which can be obtained using this method. Only three people responded in this case correctly and pointed at the classical animation, explaining that the transition from one animation to the next is too fluid and slow, which is not characteristic of realistic animation.Probably the reason why most people responded wrongly is that the use of video freference was used to create a hand animation. In the case of people choosing the right animation, most, because as many as 6 out of 8 people, responded correctly pointing to Motion Capture. This choice was mainly driven by the naturalness of

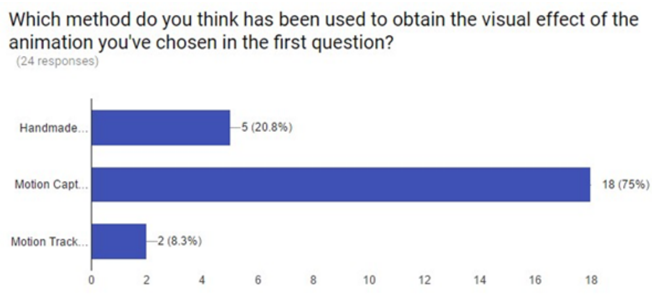


Fig. 12. Answer to third question, Paula Pszczoła, Lodz University of Technology 2017

animation and the appropriate rate with which these animations were changing. It was also pointed out that the model was not moving rigidly.

The fifth question was whether respondents were ever in touch with at least one of the three methods used (Fig. 13).

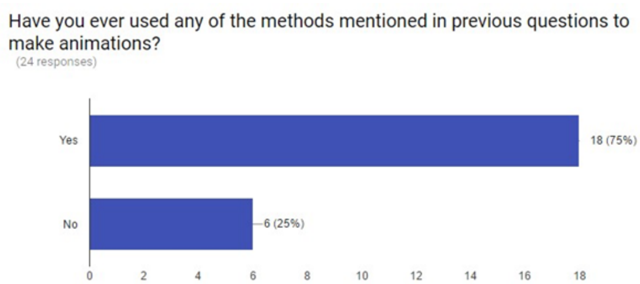


Fig. 13. Answer to fifth question, Paula Pszczoła, Lodz University of Technology 2017

The next question, the sixth, was a supplement to the fifth. If a person has chosen "Yes" in the previous one, then sixth was asked to describe their work experience with one or more of these three methods.Most of the answers show the Motion Capture method. Three people even pointed to the same software that was used in the workplace, and it was Faceware, pointing to ease of use and the results it could get. Two people have admitted to working with classical animation, claiming that animating by hand can get precise animation. No one has advocated Motion Tracking.

In question seven, respondents were to make a choice – if they would choose one of these three animation methods they would choose and why? Two out of twenty four were in favor of classical animation, supporting the arguments of the previous question. The rest of the people, unanimously stated that to get the best effects, Motion Capture is the most practical and the most accurate. It guarantees animation realism and reduces working time.

The following questions are related to the two main industries in which the Motion Capture method is quite often used, and these industries are game industries and the film industry. The eighth and ninth questions deal with the topic of creating computer games. The respondents were asked whether they had ever worked with Motion Capture in the gaming industry.

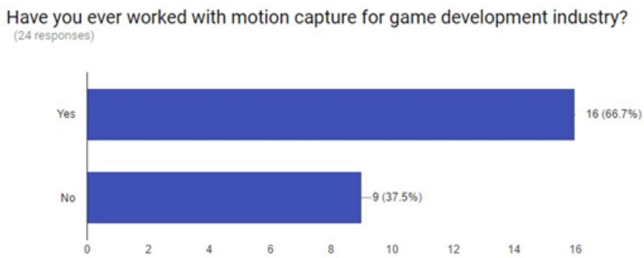


Fig. 14. Chart for answer to eighth question, Paula Pszczoła, Lodz University of Technology 2017

The ninth question was a development to the eighth (Fig. 15), where people who responded "Yes" specified what they used Motion Capture at work.

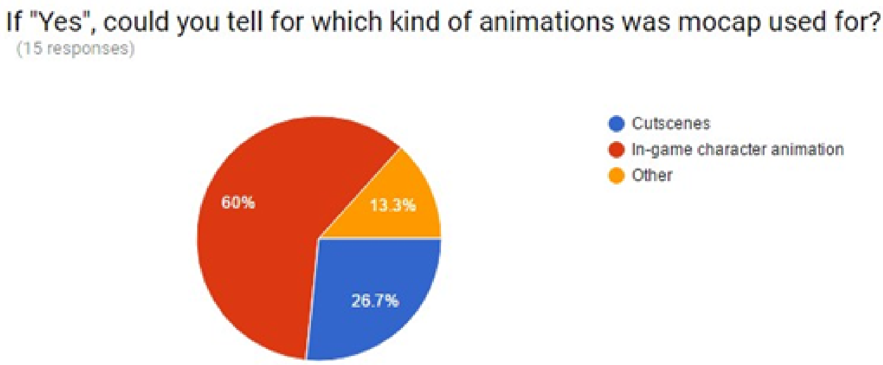


Fig. 15. Chart for answer to ninth question, Paula Pszczoła, Lodz University of Technology 2017

The study shows that, despite the fact that most of the incorrect assignment of animation to the method, Motion Capture quite clearly dominates the opinions of the respondents because of their effectiveness and received realism. Only a handful of people are more convinced of the classic animation that gives complete control over the animation. No one has advocated Motion Tracking. Probably because its more refined and more mobile version is Motion Capture.

5 Conclusions

Facial animation is a very comprehensive subject, because it involves many aspects to be considered if the animator is keen on achieving a realistic result. This work used three different facial animation methods to get the most realistic animation possible. These methods were:

- Classic animation (manual).
- Motion Tracking.
- Motion Capture.

The resulting animations were then used to conduct a survey to determine theoretically the best one.

The survey, due to its quite specialized topic, was conducted on a very small group of people. The answers obtained let us infer that Motion Capture gives the most realistic results. According to the respondents, it guarantees the best results in terms of realism and fluid animation.

The results of the research show that a well-executed classic animation can compete with motion capture animation. Keep in mind that in order to have realistic face animations, using classic animation, you should devote quite a bit of time and effort. In this respect, Motion Tracking and Motion Capture dominate the classic animation. Using Motion Tracking involves the risk of a fairly lengthy animation process if the video is not accurate or the skeleton model is not perfect. In this respect Motion Capture dominates over the previous two methods. The Faceware software that was used to make animation in the practical part is very accurate during retargeting, which largely eliminates the need to manually correct the resulting animation.

Despite having quite realistic looking animations, all of the presented methods require further testing. This is because there is no one specific use for them to create animation. For example, in our case, we used the blendshape method in each method for the animation, and as you know there are other methods of controlling the model grid. Hence the question whether the skeleton, for example, is relevant in the method used. There may be a chance that Motion Tracking is better at dealing with dinosaur-based animation than on blendshape objects. The same applies to the last two methods. Unfortunately, despite many searches, neither Blender developer nor Faceware does provide accurate information on how traffic-capture algorithms work, so there is a need for further tests with different settings for both camera and model skeleton.

REFERENCES

[1] Noh, J.-y.; Neumann, U.: *A Survey of Facial Modeling and Animation Techniques*. Integrated Media Systems Center, University of Southern California, 1998.
[2] Pushkar, J.; Tien, W.C.; Desbrun, M.; Pighin, F.:*Learning Controls for Blend Shape Based Realistic Facial Animation*. Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation, San Diego, California 2003.
[3] Terzopoulos, D.; Waters, K.: *Physically-Based Facial Modeling, Analysis and Animation*. The Journal of Visualization and Computer Animation vol. 1, 1990.
[4] Chuang, E.; Bregler, Ch.: *Performance Driven Facial Animation using Blendshape Interpolation*. Computer Science Department, 2002. Stanford University
[5] Huang, H.; Chai, J.; Tong, X.; Wu, H.T.: *Leveraging Motion Capture and 3D Scanning for High-fidelity Facial Performance Acquisition*. Journal ACM Transactions on Graphics, ACM SIGGRAPH article nr 74, 2011.
[6] B.Bickel, B.; M.Botsch, M.; R.Angst, R.; W.Matusik, W.; M.Otaduy, M.; H.Pifster, H.; M.Gross, M.: *Multi-Scale Capture of Facial Geometry and Motion*. ACM Transactions on Graphics- Proceedings of ACM SIGGRAPH article nr 33, 2007.
[7] [online] *Blender Reference Manual*. Available at: http://blender-manuali18n.readthedocs.io [Accessed May 2017].
[8] [online] *Faceware Live*. Available at: http://support.facewaretech.com/live [Accessed May 2017].
[9] [online] *Faceware Software Estimator*. Available at: http://facewaretech.com/pricing/ [Accessed May 2017].

Radosław Bednarski is an assistant professor at the Institute of Computer Science at the Faculty of Technical Physics, Informatics and Applied Mathematics at the Lodz University of Technology. He graduated in Computer Science, Computer Graphics and Multimedia Specialties. Scientific interests concern the use of modern technologies in computer games and medicine. Author of publications on the subject of multimedia and games using information technology in biomechanics.

Paula Pszczoła is a graduate of the faculty of FTiMS of the Lodz University of Technology in the field of Computer Graphics and Multimedia. Works as a 3D graphic artist in Wastelands Interactive studio in Lodz, and after hours creates own game with a group of friends. Passionate about computer games, drawing and skiing.

Adaptation of WASABI emotion engine for use in video games

Filip Wróbel, Marcin Daszuta, Dominik Szajerman, Adam Wojciechowski

Lodz University of Technology, Łódź, Poland
dominik.szajerman@p.lodz.pl

Abstract

Artificial intelligence and decision making processes were very important since the beginning of video game industry. Knowledge about it evolved and grown over the years both in terms of quantity, quality and reachable implementations. Trials to implement flexible and intelligent agents in VG were varying. Now existing AI technologies allow for creating non-player-characters (NPC) in games with advanced decision making and precise problem solving. The one reoccurring issue for virtual actors in games is believability and creating real-like behaviour. This paper tries to approach said problem by implementing WASABI emotion model and improving it through parameter classification and shouldering them on character traits, giving believable interaction with virtual actors.

Index Terms

computer games, games ai, behaviour modelling in games

1 Introduction

Video games consist of many various elements, including: rendering, physics engine, input devices or entity managing mechanisms. The last ones often use achievements of classic artificial intelligence like shortest path or decision making algorithms. Using them (classic AI solutions) often required adaptation to best suit given situation of video games. On their basis, simpler versions of finite state machines and behaviour trees were created. Mechanisms created this way served in modelling of agents in video games. Problem with them was that they require foreseeing players actions and designing adequate reactions in advance. Believability of this kind of behaviour modelling is fully dependant on quality of design and assumptions about players possible actions.

Looking for the way of improving believability of interaction, there was a search for better ways for modelling agents behaviour. One of the directions developers taken was to refer to psychological models describing behaviour, where cognition plays vital role. On their basis there were created implementations allowing for simulating decision making processes and reactions to players actions. They allowed for undertaking actions which would be perceived logic by the user.

The one still missing piece is believability. Actors behaving perfectly logic become unrealistic. And so, to resolve this issue next step is to embed emotions into the above models. This way models shouldering on logic could imitate impulsive or irrational actions in believable way. There are several models combining cognition with emotions. Problem is that they are still in prototype or research state unavailable to be used in production environment. There is a need then for analysis and adaptation of these models for the use in video games.

One of the above mentioned models is WASABI, which stands for Affect Simulation for Agents with Believable Interactivity. It presents both theoretical model and practical implementation. Problem is that given implementation is completely separate from video game production environments. This work tries to eliminate distance between WASABI model and video game environment, so it could benefit modelled behaviours believability in said conditions.

2 Related Work

Behaviour modelling will be understood as providing logic for virtual human-like actors with goal of believability. This relates to two main directions:

- Classic approach – mechanisms based on classic AI models.
- Psychological approach – mechanisms based on psychological models.

2.1 Classic behaviour modelling in games

Classic approach to video game agents behaviour modelling usually bases on one of two most popular models:

- finite state machine (FSM),
- behaviour three (BT).

Defining many singular states is core of both above models. Every single state is responsible for performing one simple action. At any given time only one state can be considered active. Differences between these two models hide in structures and state transition implementation.

2.1.1 Finite State Machine

FSM defines states as procedures with transitions contained inside the instruction list. This allows for fast creation of simple yet somewhat responsive behaviour. Problem grows with every state and transition added by rising complexity of states code. Figure 1 shows example FSM structure and flow control. This model proves itself useful in cases of smaller productions and trivial game mechanisms. Additionally it allows for constant state management where fluidity and looping of control is valued.

2.1.2 Behaviour Tree

In the opposite to FSM, Behaviour Tree separates transitions from actions. Transitions are not defined inside the state but with the use of superior structure managing flow control. Importantly these are dependant on objects parameters. Usage of this separation makes states more versatile, allowing code to be reused. It is worth noting that said superior controlling structures are often updated with different frequency than states themselves leading to higher performance and lower overhead. Figure 1 shows example Behaviour Tree structure and control flow. This model is often used agents behaviour modelling in larger productions. It is especially useful in situation where agents is required to perform series of actions or react in more complex way.

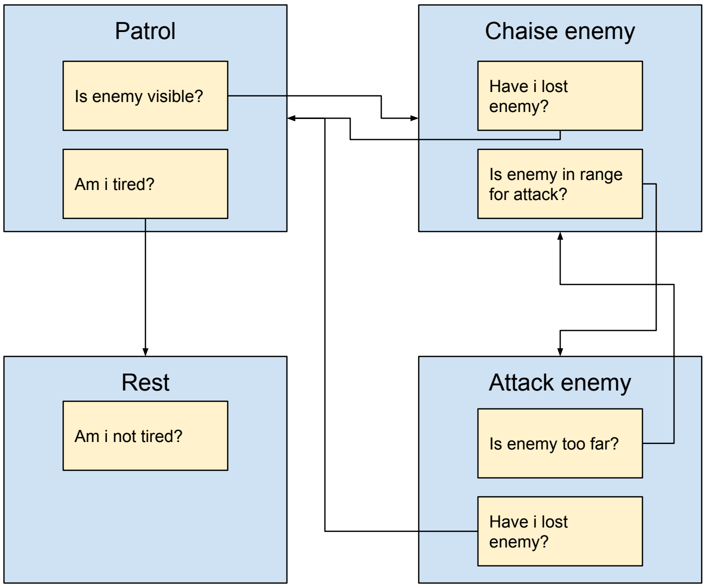


Fig. 1. Example visualization of Finite State Machine

2.2 Psychological models

Above models are classic way to implement actors behaviours. They base on scripting specific actions and predefined scenarios for player and entities. They require design and implementation of any behaviour on event-reaction rules.

Different approach is modelling agents traits. This is the place where cognition models come to play. They allow for designing not actions or condition-reaction behaviour but agents character. Cognition based agents equipped with knowledge and decision modules is capable of behaving more realistic. This shifts weight of creating believable experience from character designer to cognition module implementation.

Final step in creating more believable experience is extending cognition model with emotion simulation module. This approach ensures logic in decision making with human like element which causing rare illogical or impulsive actions cased by emotional state.

2.2.1 Cognitive approach

Modelling behaviours with cognition module extends problem of perceiving world with appraisal element. This enables for different interpretation of the same actions by different agents. It is possible because models performing decision making based on appraisal take into account their own knowledge, memory of events and performed actions.

Leading example of cognition model is BDI, which stands for Belief Desire Intention. In this model Intentions are understood as sequences of connected actions named plans. Plans can be differentiated into:

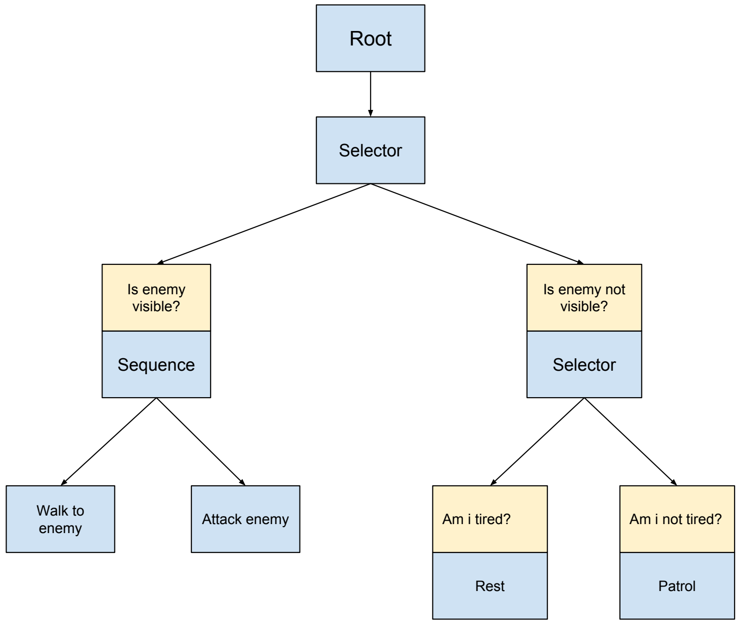


Fig. 2. Example visualization of Behaviour Tree

- “plans” – as established to perform,
- “plans as recipes” – library of actions which agent believes that translates into particular outcome.

Important to note here is that “plans as recipes” are indeed part of agents beliefs which means that they are true for this agent but could not be general truth. This can lead to interesting sequences of events, in particular when believe about particular plan is false. Agent performing action may have to adapt to situation different from the one planned, provided his belief about action results was wrong. “Plans” are actions that agents equipped with BDI module enqueues for realization. What is more, queue of these actions is then taken into account in decision making process.

Figure 3 shows flow of data and control in BDI architecture. Informations absorbed from surroundings by the agent are appraised by agents belief. This can be used to modify that way agent perceives environment. Appraised data is then source of possibilities given decision process and filters for filtering process. Similar source of possibilities and informations for Deliberation Process are Desires. These on the other hand, come from the inside of the cognition module.

This architecture of cognition module allows for great ways to base decision making on knowledge database and plans library given to the agent. This results in ability to easily modify agents character and his behaviour by changing his parameters, yet not touching library of possible actions.

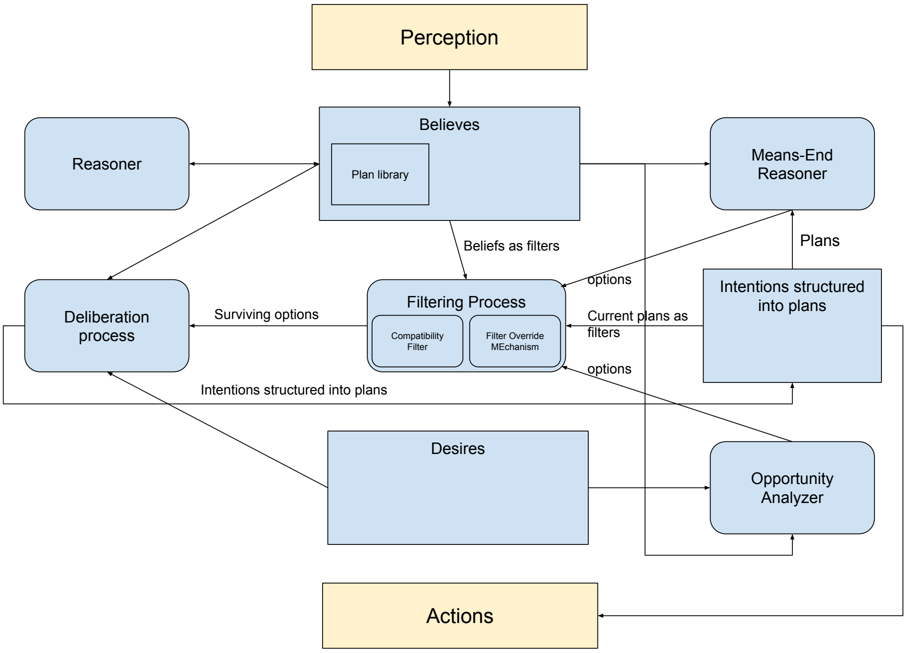


Fig. 3. BDI architecture

2.2.2 Emotion models

Emotion simulation in case of video game characters is process of generating and sharing data about emotional state. It is possible to differentiate two approaches:

- reactive – every stimulus is appraised and emotion is generated in response,
- continuous – stimuli are appraised, but they don't generate emotion immediately. Instead they change actors internal actors state, which then translates into continuous emotional state.

Example of the first approach is model OCC (Orthony, Clore and Collins) [4]. It defines tree paths for appraisal:

- Consequences of events.
- Actions of Agents.
- Aspects of Objects.

Every of the above path defines a decision tree, where evaluation for any of the above leads to generating a particular emotion in response to stimulus. This model fits any situation which requires instantaneous reaction and binary appraisal of events.

Alternative for above model is FFM (Five Factor Model) [5]. It refers to five traits of personality namely: Extroversion, Agreeableness, Conscientiousness, Emotional Stability, Sophistication. This model describes agents emotional state with use of point in space called PAD (Pleasure, Arousal, Dominance).

3 WASABI MODEL DESCRIPTION

Previous chapters presented both classic approaches to behaviour modelling and psychology based models. WASABI (Affect Simulation for Agents with Believable Interactivity) model is part of the latter group. It was created by Becker-Asano in his thesis [6] and is capable of extending cognition model with emotions, this chapter explains the basis of the model. WASABI term refers to two separate notions:

- Emotion engine.
- Architecture of connection with emotion module.

In terms if Emotion engine WASABI describes data reception, emotion state processing and emotion generation. Architecture on the other side handles the problem of connecting WASABI emotion module with exterior modules like cognition, knowledge or decision making one. It is worth noting that author of the model shared his implementation of the model through public git repository [7].

3.1 Emotion engine

This part of the model is often referred to as Emotion Engine Core. It was wisely separated into three main elements in work [8]. Highlighted there are:

- emotion dynamics,
- primary emotions,
- secondary emotions.

WASABI bases its emotion presentation on PAD space originated in Mehrabians work [5]. Because of papers focus being adaptation of WASABI for use in video games, this paper mainly describes emotions dynamics as it fits game and simulation environment best.

3.1.1 Emotion dynamics

Wasabi model comes out of assumption that emotion and mood influence each other constantly over time. This dynamic relation was digitized with the use of two dimensional space, where provided axes are responsible for simulating both short lived emotional state (named Valence) and longer lasting mood (named Mood). Third measure of actor internal state is axis responsible for measuring time period where actor is left without any stimuli (named Boredom).

All above measures build up three dimensional space called VMB, standing for Valence, Mood and Boredom. Each axis of the space is clamped to fixed ranges which are presented in the Table 1.

Emotion dynamics are designed with constant simulation in mind and so, it is important to define two time related variables:

- t – total time elapsed in simulation, can be understood to "as point in time",
- Δt – time elapsed since last update, should be of similar value over the course of simulation.

Internal state of the agents current being is represented as a point in VMB space and updated with constant frequency. The update is performed every frame of simulation separately for each of the measures with provided mathematical description. Point in VMB space in time t is defined as $VMB_t(V_t, M_t, B_t)$, where:

- V_t – Valence axis value in time t.
- M_t – Mood axis value in time t.
- B_t – Boredom axis value in time t.

Simulation of Valence axis is performed with the use of provided mathematical model model. And so WASABI defines delta Valence with equation (3) which bases on equations defining temporal values of acceleration (a_{Vt} in equation (1)) and velocity (Vel_{Vt} in equation (2)). Having calculated delta Valence it is then used to calculate final value of Valence axis with equation (4). It is important to note that although not explicitly given, but featured in implementation there is special case of zeroing Valences velocity. Whenever Valence values were to pass the neutral points (which would be result of accumulated velocity) and gain negative value due to stored velocity (Vel_V) it is rather stopped and its velocity zeroed.

$$a_{Vt} = (-vTens \cdot V_t)/mass \tag{1}$$

$$Vel_{Vt+1} = Vel_{Vt} + a_{Vt} \cdot \Delta t \tag{2}$$

$$\Delta V_t = Vel_{Vt} \cdot \Delta t + \frac{(a_{Vt} \cdot \Delta t^2)}{2} \tag{3}$$

$$V_{t+1} = V_t + \Delta V_t \tag{4}$$

Mood is simulated over time with similar mathematical model. Key difference here is influence that Valence causes to Mood, this way in every step of simulation Mood value has two delta values. First one, which reflects impact Valence has on Mood is specified with equation (5). Second one: resembling Valences model is defined with acceleration (a_{Mt} in equation (6)) and velocity (Vel_{Mt} in (7)). These are then used in calculating mass on string delta Mood in equation (8). Finally current step value for Mood is a sum of previous step Mood value with two delta Values ($\Delta M_V t$ – delta from Valence, ΔM_t – delta from mass on spring model) and is defined in relation (9). Similarly to Valence, Mood has implementation featuring very same mechanic of zeroing velocity (Vel_M). Important difference is that neutral value for Mood is not necessarily zero. Neutral point is defined as Prevalence> It may be, but does not has to, be zero.

$$\Delta M_{Vt} = slope \cdot V_t \cdot \Delta V_t \tag{5}$$

$$a_{Mt} = (-mTens \cdot M_t)/mass \tag{6}$$

$$Vel_{Mt+1} = Vel_{Mt} + a_{Mt} \cdot \Delta t \tag{7}$$

$$\Delta M_t = Vel_{Vt} \cdot \Delta t + \frac{(a_{Vt} \cdot \Delta t^2)}{2} \tag{8}$$

$$V_{t+1} = V_t + \Delta V_t + \Delta M_{Vt} \tag{9}$$

TABLE 1
Value ranges for VMB and PAD space axes

Space name	Axis name	Axis value range
PAD	[P]leasure	$\langle -100, 100 \rangle$
	[A]rousal	$\langle -100, 100 \rangle$
	[D]ominance	$\langle -100, 100 \rangle$
VMB	[V]alence	$\langle -100, 100 \rangle$
	[M]ood	$\langle -100, 100 \rangle$
	[B]oredom	$\langle -100, 0 \rangle$

With Valence and Mood defined it is then possible to present last agents state measure, namely: Boredom. It is simulated as if agent is getting bored progressively with time when his Valence and Boredom are close to neutral. Its behaviour over time is precisely described with equation (10), where ΔB in equation (10) describes change that happens every frame of simulation.

$$\Delta B_t = \begin{cases} -boredom \cdot \Delta t, \quad for \quad (|V_t| < vReg) \\ \qquad \wedge \ (|M_t - -prevalence| < mReg) \\ \\ -B_t, \qquad\qquad for \quad (|V_t| \geq vReg) \\ \qquad \vee (|M_t - -prevalence| \geq mReg) \end{cases} \tag{10}$$

$$B_{t+1} = B_t + \Delta B_t \tag{11}$$

Final in description but vital in model is way that impulses are taken and processed by engine in general. Model assumes that emotion engine is associated with cognition model and received impulses have valenced values simplified to single signed floating point number. Every time one of these is supplied model adds its value to Valence axis and nullified velocities of both Valence and Mood mass on spring models.

Above description present how internal simulation of emotion dynamics works in WASABI. Every frame WASABIs internal state is then mapped to PAD space (also clamped to specific ranges in the Table 1). For this space there are:

- P_t – Pleasure axis value in time t.
- A_t – Arousal axis value in time t.
- D_t – Dominance axis value in time t.

Mapping is performed with equations (12) and (13).

$$P(V_t, M_t) = \frac{1}{2} \cdot (V_t + M_t) \tag{12}$$

$$A(V_t, B_t) = |V_t| + B_t \tag{13}$$

Equations are provided only for Pleasure and Arousal, that is because Dominance values are derived directly from cognition module as a flat number. This is result of understanding Dominance value as agents general level of control over the situation he is in, and therefore cannot be simulated from the inside.

TABLE 2
WASABI parameters used in emotion dynamic state specification

Name	Default	Parameter description
vTens	69	spring force coefficient for V axis
mTens	10	spring force coefficient for M axis
slope	500	V and M axes relation coefficient
mass	5000	spring mass coefficient for simulated springs
vReg	5	V axis radius for perceiving value as neutral
mReg	5	M axis radius for perceiving value as neutral
boredom	50	growth coefficient for B axis
prevalence	30	neutral value for M axis

TABLE 3
Primary emotion in PAD space used in WASABI model

Emotion name	Emotion locations
Anger	(80, 80, 100)
Annoyance	(-50, 0, 100)
Boredom	(0, -80, 100)
Concentration	(0, 0, -100), (0, 0, 100)
Depression	(0, -80, -100)
Fear	(-80, 80, 100)
Happiness	(50, 0, -100), (50, 0, 100)
Friendliness	(80, 80, -100) (80,80, 100)
Sadness	(-50, 0, -100)
Surprise	(10, 80, -100), (10, 80, 100)

All of the models constants and parameters (with default values) were collected and presented with short description in the Table 2.

This summarises how WASABI engine core receives impulses, understoods them and maps its internal emotional state to well know PAD space.

3.1.2 Primary Emotions

Model defines primary emotions as points in PAD space accordingly to Mehrabians model. Their description consists of PAD space coordinate and WASABI added radius. This way relating agents emotional state from emotion engine core in PAD space to above emotion description with location and radius it is possible to tell whether emotion is active and what is the strength of the emotion. Having multiple emotions and positive radius values, agents is able to experience multiple emotion with varying strengths allowing for complex and fluent emotional state simulation.

Table 3 sums up all of the emotions used in WASABI emotional model. All of these are active whenever agents emotional state reaches part of the PAD space defined by given emotion. Only exception is surprise, which requires additional trigger from cognition module.

3.1.3 Secondary Emotions

In this emotion category WASABI introduces only three emotions:

- hope,
- fears-confirmed,
- relief.

These are taken from OCC model described in [9] in section 2. Important difference in comparison to primary emotions it that these require both particular location of emotional state in PAD space and trigger from cognition module.

3.2 Architecture of connection with emotion module

Architecture describing connection between emotion module and main module focuses mostly on data flow and general modules set-up.

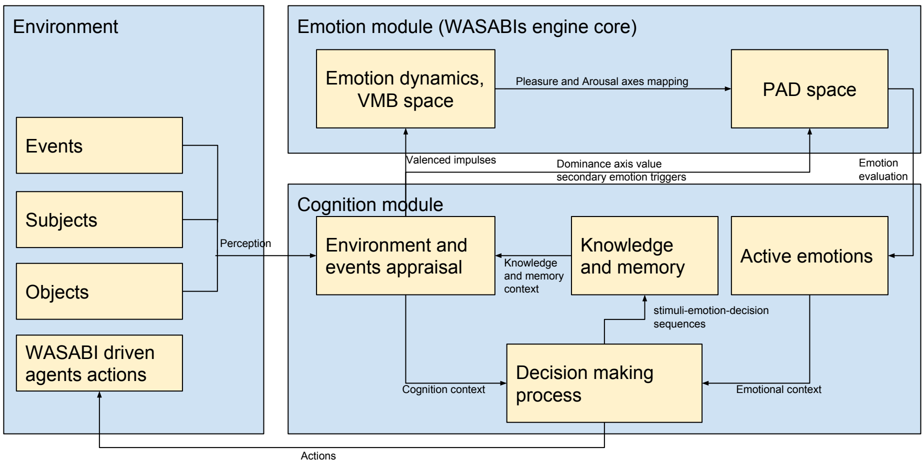


Fig. 4. WASABIs architecture of agents steering

Figure 4 shows that WASABI core in whole architecture is supporting main module in process of decision making. WASABIs author in his thesis and implementation applied BDI model. This model was chosen because of his input appraisal capabilities and ability to provide triggers for secondary emotions. It is important to note that roles of WASABIs emotional core are very limited and separate from cognition module. And so WASABI module is responsible for:

- receiving valenced impulses,
- emotional state update,
- sharing emotional state description with active emotion list.

Figure 4 show that providing already valenced impulses and usage of provided emotional state lies in responsibilities of cognition module. Also decisions and associated with them emotion are saved in modules memory for usage in future decision making.

4 WASABI MODEL ADAPTATION

Model WASABI was designed for the use of dynamic emotional state simulation, where main objective is believability. It allows or easy and reliable simulation of internal emotional state and presentation in form of PAD space point.

Despite above, said model is not perfect for video game appliances. Source of this is that parameters on which emotion dynamics shoulder are not related to character traits. From the perspective of hypothetical game designer they are just set of abstract parameters. This way even emotion based characters would behave in very similar fashion leading to exact same experience with every virtual actor driven by this model.

One models parameter that was describes as "character related aspect" in thesis [6] is "slope". It regulates impact that Valence has on Mood in internal models values. But event this aspect was only mentioned once and discontinued further in the thesis.

In order to adapt model for the use in video games it is a must to create some kind of character trait description model, that would be understandable from the standpoint of person not familiar with models internal structure.

4.1 Character traits

WASABI makes use of Mehrabians PAD space described in Five Factor Model. Relating to very same model we can find complex character description with the use of just five parameters. Each of them can be easily understood as their names are self descriptive. From the model we have:

- Extraversion,
- Agreeableness,
- Conscientiousness,
- Emotional Stability,
- Sophistication.

4.2 WASABI parameters classification with use of the FFM character traits

Main issue of this parametrization is that works related to Five Factor Model [5] [10] focuses in character traits to PAD space relation. In case of WASABI there is additional element: VMB space. In case of classic approach then we have relation: coefficients – PAD space, in this case we have: coefficients – VMB space – PAD space. Meaning that coefficients should relate to VMB space simulation and its parametrization.

Goldberg in his work [11] presented The Big-Five measures model. It enumerates 100 markers formed into 50 pairs and associates them with personality measures. These were later used by Mehrabian in [5]. This way each of the five personality factors can be described with additional 20 traits.

Each WASABI parameter from the Table 2 was assigned most accurate pair describing its influence on potential agents character. Then basing on those pairs each parameter was assigned factor from Five Factor Model associated with assigned trait pair. Resulting relations parameter – character trait pair – FFM factor were collected in the Table 4.

On the basis of created table relation between WASABI parameters and FFM elements is established. On the basis of the Table 4 and bases of WASABI emotion dynamics mapping equations were defined. Knowing default values for parameters

TABLE 4
WASABIs Emotion dynamics parameters with assigned character traits pair and FFM factors

Parameter name	Assigned traits pair	Factor from FFM
vTens	Careless-thorough	Conscientiousness
mTens	Careless-thorough	Conscientiousness
slope	Suggestive-Independent	Emotional-Stability
mass	Careless-thorough	Conscientiousness
vReg	Imperceptive-perceptive	Sophistication
mReg	Suggestive-Independent	Sophistication
boredom	Suggestive-Independent	Sophistication
prevalence	Apathetic-enthusiastic	Extraversion

it is possible to calculate value specific for any given character personality defined with FFM personality traits. For this goal there were defined:
x_0 – default value,
x_c – value calculated as specified for particular character set-up,
where: $x \in \{ \begin{matrix} vTens, mTens, slope, mass, \\ vReg, mReg, boredom, prevalence \end{matrix} \}$.

Defaults bases on original values of WASABIs model and its implementation from Becker-Asanos git repository. In order to define mapping equations there were defined:
Extra – coefficient for character trait Extraversion,
Agree – coefficient for character trait Agreeableness,
Conc – coefficient for character trait Conscientiousness,
Stab – coefficient for character trait Emotional-stability,
Soph – coefficient for character trait Sophistication.
Each of the above character traits can posses value in $< -1, 1 >$ range. With above Five Factor Model character factors there were presented mapping equations.

$$vTens_c(Conc) = vTens_0 \cdot (1 + 0.25 \cdot Conc) \tag{14}$$

$$mTens_c(Conc) = mTens_0 \cdot (1 + 0.25 \cdot Conc) \tag{15}$$

$$slope_c(Stab) = slope_0 \cdot (1 - 0.72 \cdot Stab) \tag{16}$$

$$mass_c(Conc) = mass_0 \cdot (1 + 0.25 \cdot Conc) \tag{17}$$

$$vReg_c(Soph) = vReg_0 \cdot (1 - 0.42 \cdot Soph) \tag{18}$$

$$mReg_c(Soph) = mReg_0 \cdot (1 - 0.42 \cdot Soph) \tag{19}$$

$$boredom_c(Soph) = boredom_0 \cdot (1 - 0.42 \cdot Soph) \tag{20}$$

$$prevalence_c(Extra) = prevalence_0 \cdot (0.19 - 0.81 \cdot Extra) \tag{21}$$

Constants present in equations 14 15 16 17 18 19 20 21 are reflection of impact that any of the given character factor does on the WASABI parameter. Values were taken from Mehrabians paper [5], where they were presented as amount of Reliable Variance for each of the factors in the mentioned source.

Above work proposes parametrization for WASABI coefficients with use of FFM character traits that allows for usage in video games.

4.3 Architecture

Accordingly to works technical assumptions implementation shoulders on Unreal Engine 4. It is complex game development production used engine which bases on C++ programming language and inside graphical scripting language. This scripting language is named Blueprints. Language C++ used in this engine is extended with custom reflection system based on macros which allows for usage of almost all c++ created entities in Blueprints.

It was decided to create emotion engine in C++ in way that removing all custom macros would yield clean c++ code granting reusable code for other appliances.

Implementation bases its core rules on WASABI model architecture, where emotion dynamic, cognition and memory modules are separate entities. All of them were packed into one managing class that is responsible for receiving impulses, updating and presenting emotional state. Relation between classes are presented in the Figure 5.

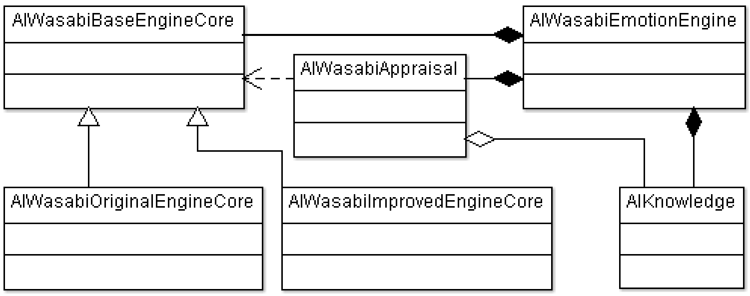


Fig. 5. Architecture of test environment implementation

As mentioned above these is one main class namely: AIWasabiEmotionEngine. It is only externally visible object. For the use of comparison and proving classification valid there was created polymorphic relation. And so AIWasabiBaseEngineCore is base class proving interface for taking valenced impulses and sharing its emotional state. Derived classes:

- AIWasabiOriginalEngineCore – original implementation of WASABIs emotion dynamics,
- AIWasabiImprovedEngineCore – implementation of emotions dynamics incorporating parametrization.

Usage of inheritance mechanisms allowed for easy comparing and switching emotion dynamics implementations and entities for potentially different emotion dynamics implementation and parameter values.

Because this work focuses on emotion dynamics module, cognition (AIWasabi-Appraisal in the Fig. 5) and knowledge (AIKnowledge in the Fig. 5) modules were created in minimalistic way containing only features required for precise emotion dynamics testing. This way external stimuli are directly translated info valenced impulses for WASABIs emotion engine core and resulting emotional state is directly passed to external logging tool. Finally dominance value described in WASABI model description is passed as flat set value, since tests are to be performed in closed and strict environment.

5 TESTS AND ANALYSIS

Earlier chapters presented WASABI model and his parametrized version. Parametrized version allows for simulating agents emotional state in game development environment. In order to test and analyse proposed solution there is a need for:

- creation of test environment,
- test and analysis of model parametrization.

5.1 Test environment

In order to create proper test environment there were created following criteria:

- controlled number of stimuli,
- repeatability and cyclic character,
- embedding test environment in engine mechanisms,
- clarity.

Basing on the above it was decided to create test scene with emotion module equipped actors. Actors will move on a loop around the simple scene, while stimuli generating object will be placed around the path. Every stimuli generating object will have either positive or negative value which in actors perception will be passed to emotion module. It is important to note that simulation will consist of multiple agents which are simulated independently from one another.

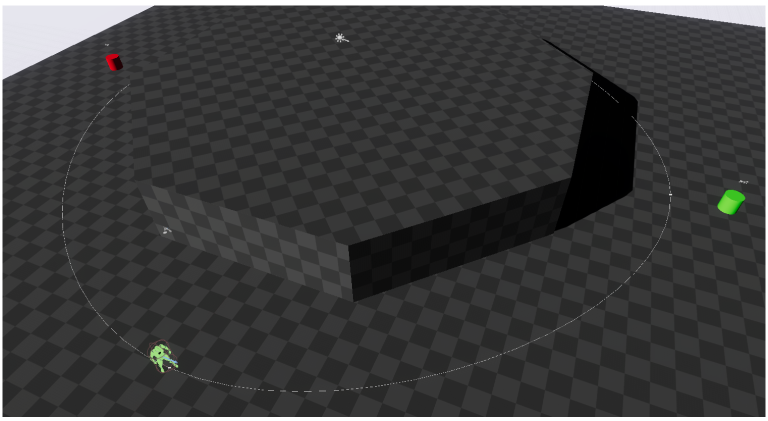


Fig. 6. Example test scene

In the Figure 6 it is possible to perceive main scene objects:

- mannequin, which is embodiment of emotion agent,
- white line, which shows the route of the actor,
- perceiving green of red object by AIPerception causes passing its perceived objects data to cognition module and further to emotion engine.

At any given point on scene there can be any amount of actors, because their potential mutual collisions were disabled. This way every emotion test pass is regulated by modification of count, placement and internal values of stimuli objects.

5.2 Parametrization analysis

In order to prove correctness of general direction of parametrization there were performed comparison test between actor equipped with same emotion engines but parametrized with different character trait values. Test can prove whether changing trait values will yield expected result. For this goal test scene was set-up in a way presented in the Figure 7.

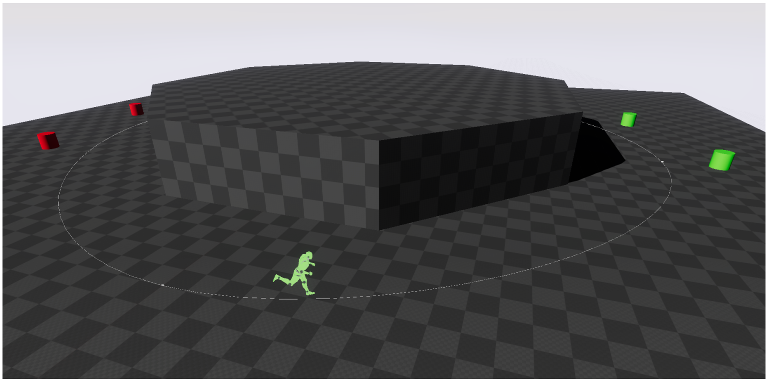


Fig. 7. Test scene adjusted to match test case

Because of broad possibilities of character creation and infinite variants of character trait it was decided to create two test cases. Each of them would test influence of single character trait parameter has on emotion dynamics and generated emotions.

On basis of the above there were chosen:

- Extraversion,
- Emotional Stability.

For each of the test cases there were created two characters of which one is parametrized with high value (positive) of given trait and the other with low value (negative) of the same trait.

5.2.1 Analysis of Emotional Stability as parameter

First step in order to analyse influence of the titular parameter is to define expected behaviour. For this goal work [12] was referenced, which characterizes among others FFM parameters. These is no direct description of Emotional Stability, but is complex description of Neuroticism, with note that it is direct opposite of the Emotional Stability from FFM. In that work, characters with high level of Neuroticism are presented as anxious, unstable and impulsive. Referring it to parameters of simulated emotional state it can be assumed that virtual actor with high level of Emotional Stability will be stable, concentrated, levelled and non-impulsive. Parametrization actor with low (negative) value of Emotional Stability will result in simulation of emotional state close to Neurotic actor character described in [12].

Shouldering on the above it was decided to choose emotion pair serving the role of analogy to above description. This way strengths of chosen emotions would have been able confirm or deny correctness of given parameter appliance in emotion dynamic parametrization. Another thing to note is that emotion have to be chosen from ones implemented in WASABI model. And so chosen emotions were:

- Concentration,
- Surprise.

With chosen emotions and prepared character trait set-ups simulation was performed. Data from simulation was collected to file and presented on charts 8 and 9, which shows both emotions strength over time with marked stimuli. It is visible that for agent with positive Emotional Stability, emotion Concentrations strength rises faster and to greater values than for agent with negative Emotion Stability whenever there is a pause in stimuli experiencing. Additional it is worth noting that value of Surprised emotion shows greater values in reaction to stimuli in case of agent with negative Emotional Stability.

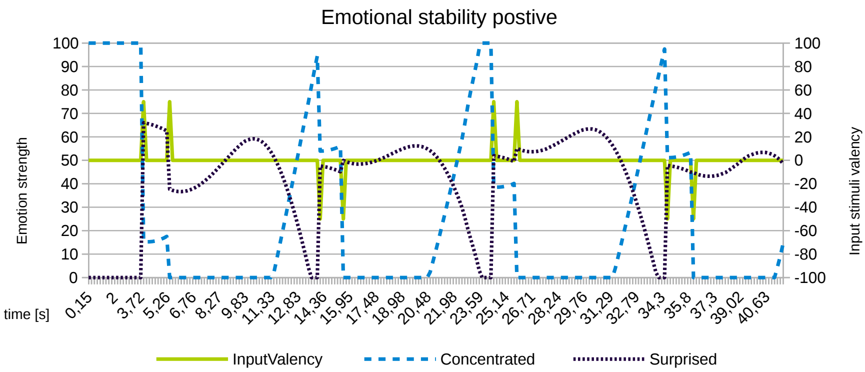


Fig. 8. Emotion strengths over time for actor with positive value of Emotional Stability

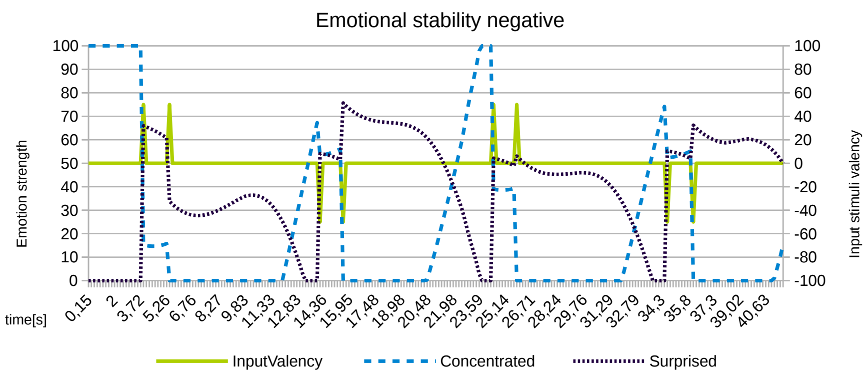


Fig. 9. Emotion strengths over time for actor with negative value of Emotional Stability

Above described behaviour and differences between prepared agents characters matched expected behaviour leading to conclusion that Emotional Stability parameter is correct in terms of behaviour modification and its direction.

5.2.2 Analysis of Extraversion as parameter

In analogy to previous sub chapter it is important to describe expected influence of parameter on emotion dynamics. Again referring to [12] agent with high values of Extraversion was described as enthusiastic or outgoing. It is possible to reason that increasing Extraversion parameter should ease of achieving high levels of happiness, joy or friendliness. In the same time decreasing Extraversion parameter value should cause emotions like sad, fear or distress to appear with greater strength.

On the basis of the above there were chosen emotions. This time because of WASABIs emotions being defined closely there were chosen two pairs of emotions:

- Happiness – sadness,
- Friendliness – fear.

Similarly, this time prepared agents were parametrized with one parameter, namely Extraversion. Created agents were:

- agent with high Extraversion value (positive),
- agent with low Extraversion value (negative).

For both agents there were conducted tests and data collected. This data was then used to create charts: 10, 11, 12 and 13.

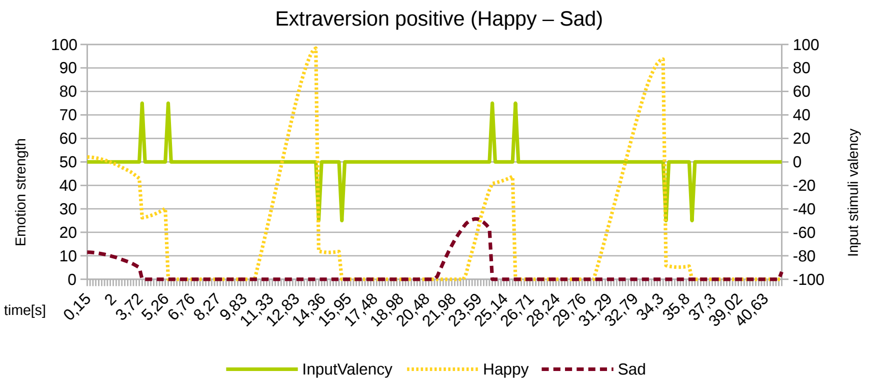


Fig. 10. Emotion strengths over time for actor with positive value of Extraversion with emotions Happy and Sad

Charts 10, 11 focus on emotion pair Happy-Sad, they shows that whenever emotion Happy is dominant at any point it is simulated with greater strength by agent with positive Extraversion. With no surprise whenever emotion sad was dominant it was experienced with higher strength by agent with low value of Extraversion.

For the friendliness – fear emotion pair there were created 12 and 13 charts. With these it is possible to notice that emotion Friendly was experienced with greater top value and through longer time by agent with character that was parametrized with high Extraversion in comparison to agent parametrized with low Extraversion. Respectively Fear was being experienced stronger by agent with low Extraversion trait parameter.

Finally it is possible to state that agent whose emotion engine was parametrized with high Extraversion showed tendency to experience emotions like happy or friendly with more ease and to greater value. Negative Extraversion on the other hand was casing actor to simulate emotions like sad or fear with greater strengths.

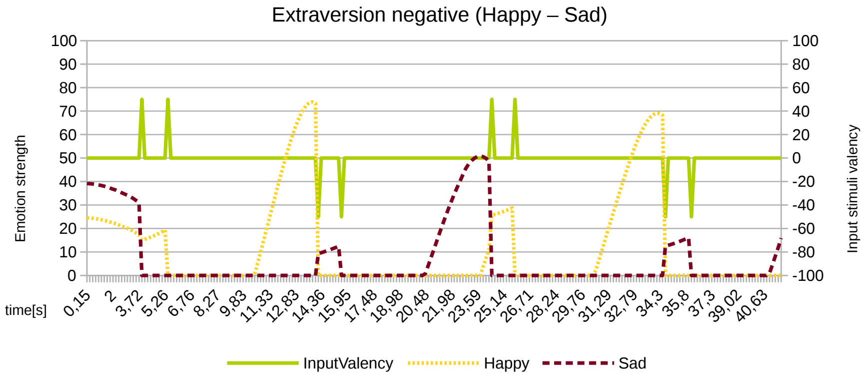


Fig. 11. Emotion strengths over time for actor with negative value of Extraversion with emotions Happy and Sad

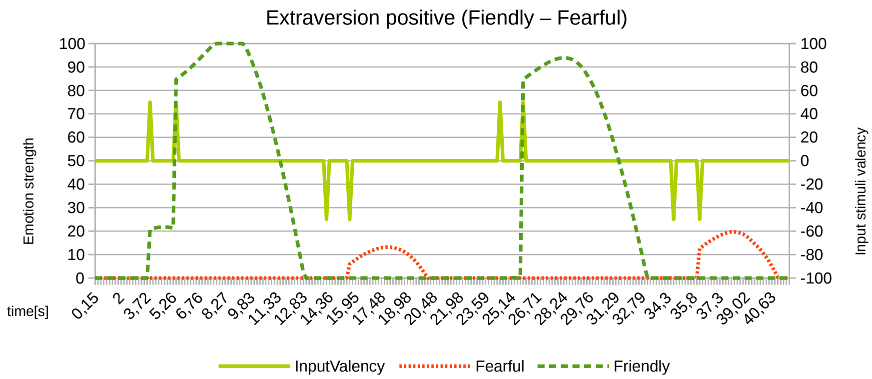


Fig. 12. Emotion strengths over time for actor with positive value of Extraversion with emotions Friendly and Fearful

Above comparison results, both for first and second pair brought expected results and confirmed proposed parametrization for Extraversion parameter.

6 Conclusions

Goal of this paper was to adapt WASABI emotion model for the use in video games in role of increasing agents believability. In order to do so there was conducted analysis

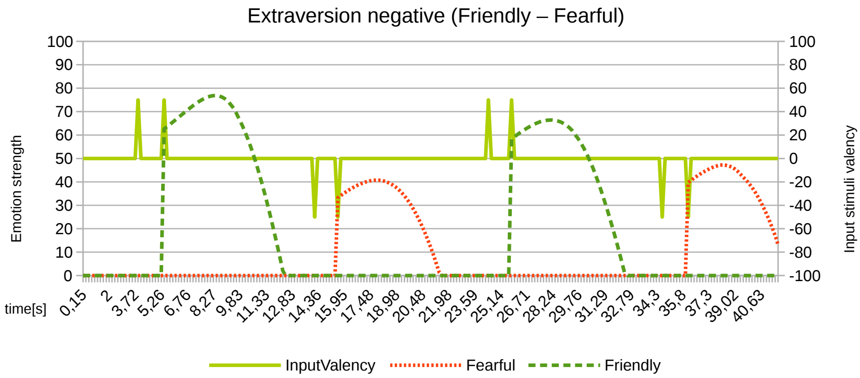


Fig. 13. Emotion strengths over time for actor with negative value of Extraversion with emotions Friendly and Fearful

of classic approaches to agent behaviour modelling and description of some of the available psychological models. Then there complex description of WASABI models emotion dynamics, which was followed by problem diagnosis and proposed solution. Proposed solution was then tested with the use of:

- implementation in video game engine,
- closed test environment.

With the use of these elements conducted were test which yielded emotional state description data. Analysis of this data lead to conclusion that proposed solution is correct on the tested field in terms of direction. Finally it is possible to state that the results were promising and allowing for future research and improvements on the matter.

Acknowledgment

This work was supported by The National Centre for Research and Development within the project "From Robots to Humans: Innovative affective AI system for FPS and TPS games with dynamically regulated psychological aspects of human behaviour" (POIR.01.02.00-00-0133/16).

References

[1] Geddes, K.O.; Czapor, S.R.; Labahn, G.: *Algorithms for Computer Algebra*. Kluwer, Boston, 1992

[2] Hamburger, C.: *Quasimonotonicity, regularity and duality for nonlinear systems of partial differential equations*. Ann. Mat. Pura. Appl. 169, 321–354, 1995.

[3] *C++ reference*. http://en.cppreference.com/, 20 Aug. 2016.

[4] Kessler, H.; Festini, A.; Traue, H.C.; Filipic, S.; Weber, M.; Hoffmann, H.: *SIMPLEX – Simulation of Personal Emotion Experience – A new Model of Artificial Emotions*. ISBN: 978-3-902613-23-3, University of Ulm, Medical Psychology, Institute for Media Computing, Germany, 2008.

[5] Mehrabian, A.: *Analysis of the Big-five Personality Factors in Terms of the PAD Temperament Model*. Australian Journal of Phychology, 1996.

[6] Becker-Asano, Ch.: *WASABI: Affect Simulation for Agents with Believable Interactivity*. IOS Press, 2008.

[7] *CBA2011/WASABIEngine: WASABI Affect Simulation library*. https://github.com/CBA2011/WASABIEngine, 11 Sep. 2017.
[8] Sánchez L.P.: *Gaming with Emotions: An Architecture for the Development of Mood-Driven Characters in Video Games*. CETINIA, 2013.
[9] Ortony, A.; Clore, G.; Collins, A.: *The cognitive structure of emotions*. Cambridge University Press, Cambridge, MA, 1988.
[10] Mehrabian, A.: *Relationships Among Three General Approaches to Personality Description*. The Journal of Psychology: Interdisciplinary and Applied, 1995.
[11] Goldberg, L.R.: *The Development of Markers for the Big-Five Factor Structure*. Psychological Assessment, 1992.
[12] McMrae, R.R.; John, O.P.: An Introduction to the Five-Factor Model and Its Applications. Journal of Personality, 1992.

Filip Wróbel Master of Science – graduate of first and second degree Computer Science on FTIMS division from Łódź University of Technology in "Games Technology and Computer Simulations specialization" and "Games Technology and Interactive Systems" respectively. Participant in UE founded project: "Informatyka – kierunek zamawiany na Politechnice Łódzkiej" in module: "Grafika Komputerowa w Rozrywce i Grach Komputerowych". Four times scholar of Lodz University of Technology rector scholarship.

Marcin Daszuta Ms graduate of the second degree studies in Information Technology at the Faculty of Technical Physics, Information Technology and Applied Mathematics at Lodz University of Technology in specialization Computer Simulation and Games Technology and Ph.D. student at the same Faculty. Laureate of ZTGK tornament in 2013 and participant of Imagine Cup competition in 2015. Co-author of a few scientific publications. Participant of scientific research project "From robot to human, innovative technology allowing to fluent and dynamic regulation of influence cognitive and emotional psychology on actions of AI" as developer. Employee in Ericsson Ltd. company as software developer for 2 years. Interested in computer games, computer grathics, virtual reality, computer stimulations, mobile systems, mobile applications, artificial intelligence, level design in computer games.

Dominik Szajerman was born in Łódź, Poland, in 1977. He graduated in computer graphics and obtained the Ph.D. degree from Department of Technical Physics, Information Technology and Applied Mathematics at Technical University of Lodz, Poland, in 2001 and 2010, respectively. During his pre- and postgraduate experience he was involved in a number of research projects in the areas of computer graphics, computer games and human-computer interaction. His academic interests lie in the computer graphics, GPU processing, computer games development and image processing.

Adam Wojciechowski PhD, DSc, Institute of Information Technology, Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology.

Supervisor of an IT teaching specialisation on Computer Games and Simulations. Since 2016 vice-Dean for Development and Business Cooperation at the Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology. Since 2016 vice-chairman of the GameInn Steering Committee in the National Centre for Research and Development. The Lodz University Rector representative in the Lodz Game Dev Cluster and Faculty Dean representative in the ICT Central Poland Cluster.

Research interests encompass: human-computer interaction, cognitive computing, computer graphics, computer vision, computer games, virtual environments, computer simulations and visualizations, computer animations, biometry and machine learning. Author or co-author of over 50 research papers. Coordinator of 3 and participant of 13 national and international research and development projects, financed by Polish and European programs.

360 degree movie as novel method for storytelling by image

Karolina Nowak, Rafał Szrajber

Lodz University of Technology, Łódź, Poland
karolina.nowak@dokt.p.lodz.pl, rafal.szrajber@p.lodz.pl

Abstract

Immersive video is getting more and more popular as an instrument for canvassing image. It is a novel area of research on space representation, possibility of receipt space, interaction methods and forming visual announcement. The new phenomenon supported by technologies is implementing variant ways of telling the story by images. From that forward, a frame is the whole field around spherical camera and only spectators are able to decide what currently is being displayed. The frame is not a method of image composition formation anymore, so it involves pioneering researches and trailblazing forming video's content. This article is an introduction for understanding major distinctions between narrating history with classical film and 360 degree video. Furthermore, on the grounds of our own video realization, it is also showing potential advantages and disadvantages of this film genre.

Index Terms

360 degree movie, spherical movie, interactive film, virtual reality, computer graphics

1 Introduction

First panoramas came into being in the end of baroque (XVII/XVIII century). In 1787 Robert Baker patented his idea of panorama painting creation and exposition. He painted sequence of pictures and connected them into one 360 degree painting. He also elaborated the plans of rotunda, the building presenting panorama image. The paintings stood next to round walls and spectators were in the center of construction. Phenomenon of panorama can be noted in photography too. First panoramic camera with 150 degree field of view was invented in 1843. Camera lens was moving by hoisting winch while light-sensitive material was in stillness. First 360 degree camera was developed in 1857 in England. After almost 50 years there was first open-access professional panoramic camera – Cirkut. Raoul Grimoin-Sanson patented a cinéorama in 1897. Three years later he presented it on the worldwide expo in Paris (France). Spectators stood on the platform looking like a basket of a balloon and below them there were 10 projectors displaying 360 degree movie. Film was recorded by 10 cameras attached to the flying balloon over Paris. It was the first and the last attempt of this type of movie for a long time. Cinéorama was closed after 3 days because of security reasons.

In 1953 the first movie in panoramic process CinemaScope was realized. The image was two times wider than the standard one. Using this effect succeed so late because only 20th Century Fox studio found practical application for special lens invented in 1926. Circarama appeared in 1955 in Disneyland theme park. Its working idea was similar to the cinéorama. It used 9 projectors that were mounted between screens standing in the circle. The image was displaying through the whole space for the spectators. Films were recorded by 9 cameras connected with each other. Success of circarama gave the incentive to create subsequent constructions for presenting 360 degree movies. Circular kinopanorama was built in Moscow in 1959 and after 4 years circlorama was established in London. They have 11 projectors.

Nowadays, the 360 degree movie is getting more and more popular. It is a spherical film presenting the whole, boundless space around camera. Field of view equals 360 degree and every part of the environment is recorded in the same time. Spectator can decide which film scrap is displayed so it gives opportunity for watching the same movie in many different ways. There are two the most frequently used projection modes: global and panoramic. Understanding of spherical films' changing process allows us to define modern image of storytelling by 360 degree movie. This topic is a new research area between classical film and virtual reality so we want to introduce major distinctions between narrating history with 2D film and spherical movie.

2 Materials and methods

The main purpose of conducted works is review and realization comparative materials in two thematic areas:

- Creative process that is the part of each stage of making spherical movie such as preproduction, production and postproduction.
- Spectator's reaction to 360 degree movie and building recipient's experience.
- Comparison of these materials with classical 2D films brings analyze of storytelling by the image in two different ways.

2.1 Preproduction

Preproduction is a crucial element in process of filming story cultivation. It defines main presuppositions and the way of final image presentation. Our researches concern notation of storyboard as the first material visualizing not only shot action, but also camera motion and image contents.

We decided to prepare four movies – two stories presented in classical 2D and 360 degree films. The first one shows sequences of actors' motions such as pacing through the corridor and entering or exiting hotel rooms. The second one puts on hide-and-seek game. Actors are walking round in the forest, the woman is hiding behind trees and the man is looking for her. Their motions are simultaneous. Considering our ideas about movies' plot, we wrote short scenarios describing when and in what way each actor has to move about. It is very important in both films as this element is supposed to accentuate distinct differences between classical 2D and 360 degree movies. Next, we prepared storyboards based on scenarios. We applied different types of shot actions from master plan to close-up to avoid spectators getting bored by unvarying image. Actors' figures are simplify so we had to add sex symbol in some cases. It is a popular way of planning movie image in 2D film, but spherical movie pictures 360 degree scene so this process is a little pointless there. We decided to draw walking plan for actors what is a sequence of their consecutive motions. The man and woman are distinguished by arrow colors showing where they should go. The black circle in the center of scene is a symbol of camera that is placed on floor or attached to ceiling. In Figure 1 we compared these two types of storyboard, one for 2D and one for 360 degree movie.

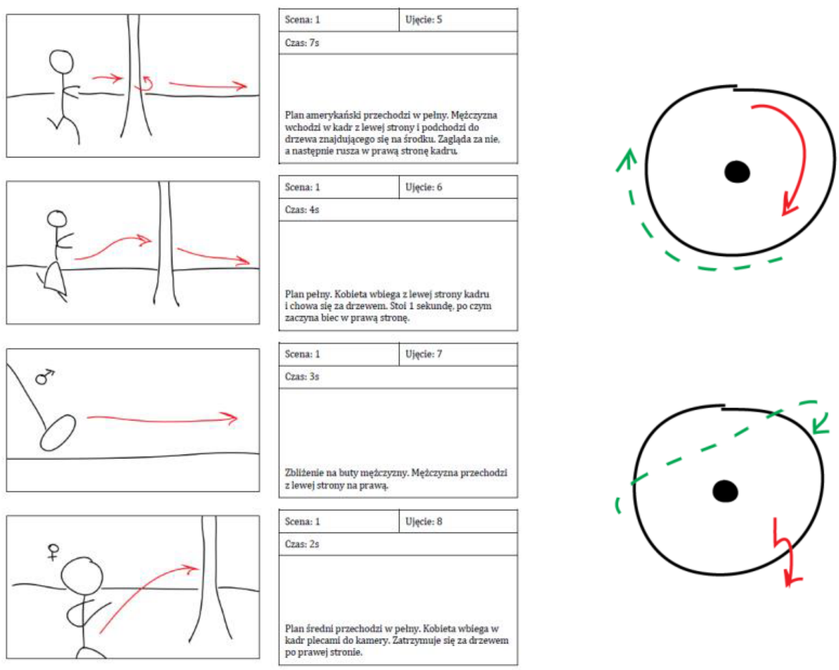


Fig. 1. Comparison of 4 scenes in two types of storyboard. The first one (on the left) is a classical version for 2D films and the second one (on the right) is a walking plan for actors. The dotted arrows represent the woman and the solid ones – the man. Source: Authors'

2.2 Production

In this section we focused attention on some research areas:

- Shaping first shot (shot actions, dynamics, etc.) as an element beginning the whole film story.
- The way of shifting shot actions when place or space observer point is changed.
- Definition of main plot and communication with spectators towards film legibility in regard to unfettered image exploration.

We made four movies as a result of preproduction works and one coverage of Zespolowe Tworzenie Gier Komputerowych competition as unplanned 360 degree documentation. We used Kodak PixPro SP360 camera for spherical movies (HD resolution) and Nikon Coolpix S7000 camera (1920x1080 pixels resolution) for classical ones. We got films in MP4 and MOV formats as they are the most popular ones for being displayed nowadays.

Both spherical films with scenarios and walking plans have one shot from the beginning to the end of movies. Camera was mounted on trivet on the floor or was applied to the ceiling. Actors were walking sequentially or simultaneously depending on our earlier ideas. We made efforts to get environment recognizably as trees in forest or doors in hotel corridor are very similar to each other. For example, in the first end of the hall there were vacuum cleaner and trolley with cleaning accessories while the second one ended with the window. We assumed that spectator would be able to differ surroundings then and explore the image freely.

We wanted documentation of competition to make spectators feel like they were participants of this event. We mounted camera on a trivet and put it among the people. It was on the height of sitting people in auditorium and recorded the image during finals. We got screenshots with games and reaction of people playing these games contemporaneously as camera stood between participants and screens.

This documentation shows many episodes during competition event so changing images in movie was extremely necessary. The main problem is how to connect these shots when spectators can watch the part of the scene that they want to. We decided to use sharp cut that is generally applied in classical 2D movies. The main action of one shot covered the one of previous shot so the major point of scene is always in the same place of image.

2.3 Postproduction

Postproduction is the most important process that weighs in final movie image directly. We completed spherical film with additional elements and get comparison of work techniques and presentation methods for classical and 360 degree movies.

2.3.1 Methods of working with a spherical image

There are many desktop applications provided by company producers of 360 degree cameras. It facilitates basic functionalities such as changing projection mode of spherical films, cropping movies or saving one film frame as image. However, 360 degree movie is still a movie so widely known programs for processing films can be used. Adobe offers applications that are sufficient for these types of work. Adobe After Effects adds special effects, gets better film quality and turns movie global mode into panoramic one and vice versa. Adobe Photoshop is good for making graphics that are going to be stowed in the film. Adobe Premiere Pro puts layers of image and sound on each other and cuts scenes and merges them.

Our 360 degree movies have form of 1440×1440 pixels square. Recorded environment is presented as a circle with 720 pixels radius. Empty spaces in the box corners are black. Unfortunately, film saved in this way is very hard to be processed and impossible to be played in popular applications. We had to change its projection mode in Adobe After Effects program. We used Polar Coordinates function as it transforms circle image into square one and vice versa. It shifts round of pixels with bigger and bigger radius into increasingly lower row of pixels in square image. This process is realized until radius gets its greatest value: $720\sqrt{2} \approx 1019$.

Fig. 2. Panoramic shots from 360 degree movies. From the top: actors' changing rooms in hotel corridor, hide-and-seek game in the forest, documentation of Zespolowe Tworzenie Gier Komputerowych competition: camera in the center of final's audience and between the screens displaying game and its players. Source: Authors'

Examined circle starts in -180 degree angle and pixels are relocated counterclockwise. Complete with increasing radius, the interpolated colors of shifted pixels turn down as more and more data are signed over. When the radius of pixels' circle is bigger than 720, the algorithm has to save empty fields as invisible ones with alpha level in the final image. At the end of this function process, the image is scaled by its height.

Invisible pixels with alpha level in panoramic mode can be masked by special prepared graphic. It can be one consistent color or can present some logo, text or image. As movie is carried over the sphere during being displayed, panoramic image at the top and bottom gets modified significantly. The rectangle strip turns into circle. In the picture the "circle" has the format 2:1 as a standard of panoramic image so it is an ellipse at bottom. This version of graphic is prepared in polar coordinates, so it has to be converted into rectangular one.

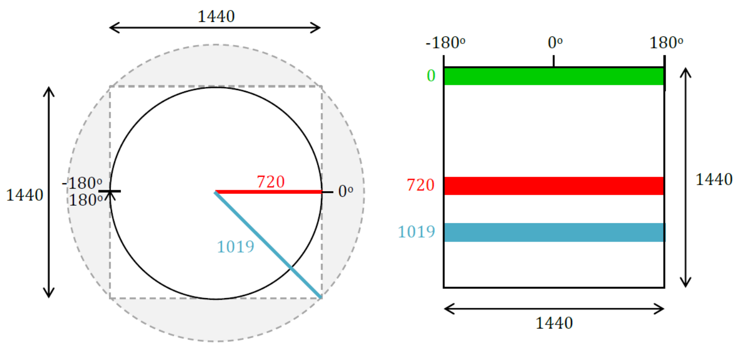


Fig. 3. On the left: input image in global mode. Smaller circle (radius equals 720) marks recorded environment, square is a saved form of movie, bigger gray circle (radius equals 1019) means pixels not existing in global mode but reflecting in panoramic one. On the right: square image consists of transformed pixels' circles. Numbers on the left are consecutive values of radius. Source: Authors'

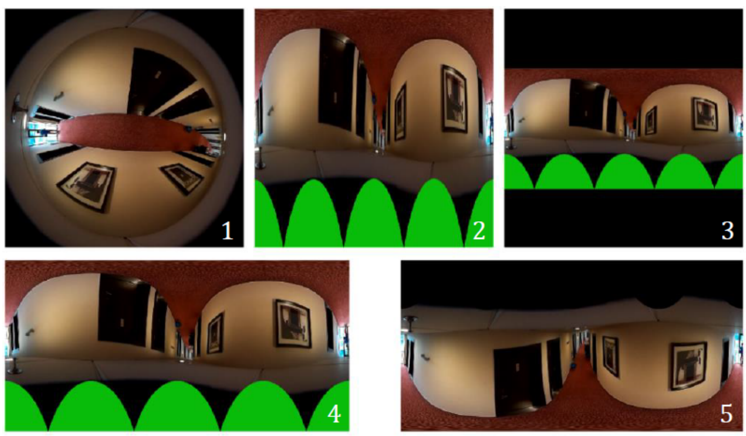


Fig. 4. Consecutive stages of changing movie global mode into panoramic one. Gray color on the bottom in pictures 2-4 marks invisible pixels that are not saved in the input image. Source: Authors'

Captions for 360 degree movies are very similar to the classical ones for 2D films. The only difference is their repetition. The text has to be written at least 3 times and disposed in equal distance so that spectators can read it without rotation of their head. What is more, this type of captions is always seen on the screen so it is impossible that someone misses them unknowingly.

2.3.2 Methods for presenting and sharing a spherical film

Some mobile applications for 360 degree cameras make live preview accessible during recording the film. There are many functionalities such as changing projection

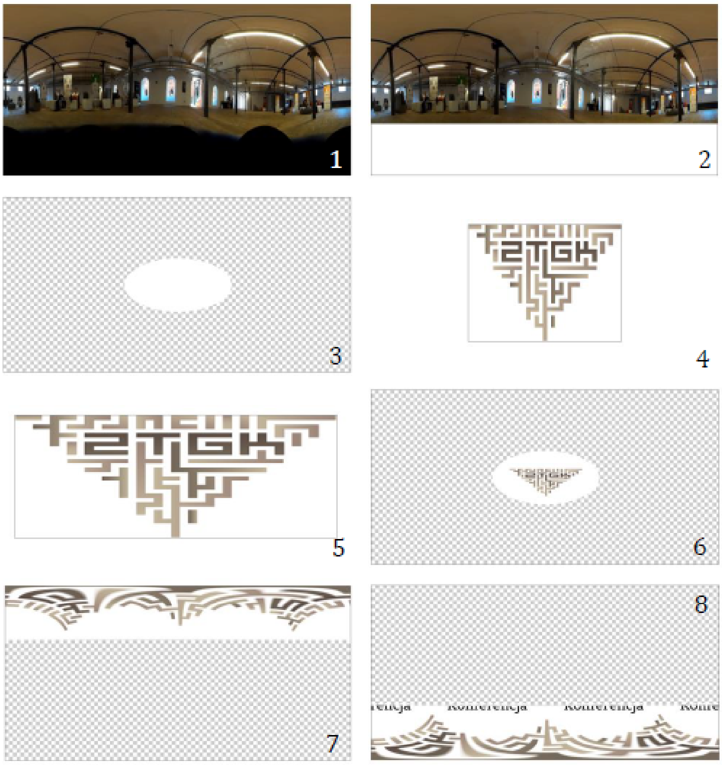


Fig. 5. Consecutive stages of preparing special graphic masked invisible pixels in panoramic mode movie image. Source: Authors'

mode, applying HDR function, disposing exposure and white balance. The list of movies is available and can be displayed on mobile or tablet screen. Starting and stopping the process of recording film are the most important and the most useful functions so there is no need for anybody to be in camera's field of view.

Nowadays, 360 degree movies are widely known and watching them are extremely easy. There are many online services with special playback like YouTube. Its interface has usually additional buttons: navigation allowing movie rotation and miniature of visible part of the film. Other option for watching spherical films is a special virtual reality device looking like a helmet. It consists of display it uses dedicated smartphones. Their price varies a lot however all of them are open-access. The cheapest is Google cardboard, but Samsung Gear VR, Oculus Rift and HTC Vive are not less popular in VR industry.

We carried out research about 360 degree movies in the group of 11 people (20-30 years old, 2 women and 9 men). We made individual, in-depth interview – research was partly managed basing on scenario. The survey consists of 4 parts, each of them started with projection of one film (2D or spherical one). Then, respondents had to answer some open questions about what they had already seen. As a movie player

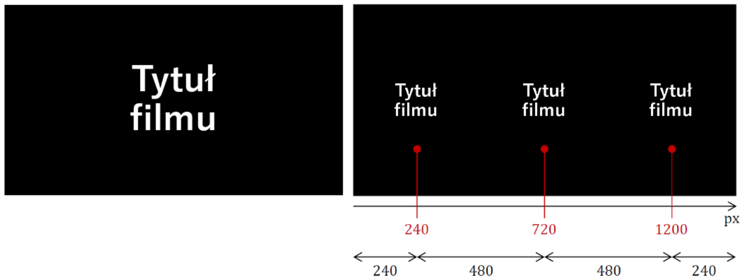


Fig. 6. Comparison of captions for classical 2D film (left one) and 360 degree movie (right one). On the right the text is repeated three times and the space between them is equal. Upper value means the center of one text block while the arrow below – the distance between them. Source: Authors'

device, we used Lenovo Yoga 2 tablet and 360 VR Player | Videos application made by Buzzard company.

3 Discussion

Majority of respondents (9 of 11 people) did not recognize 360 degree movie when it was played without information about type and features of this film. All of participants had come in contact with this technology earlier but neither numerous actors' waking out of film frame nor sounds of opening and closing doors encouraged anybody to turn the movie. These respondents were watching curved ceiling, walls and floor for 3 minutes. However, two participants recognized spherical film by dint of cardboard icon in the top-right corner of player application and swiveled the movie image freely.

Respondents were asked to draw the walking plan for the actors after seeing the classical and 360 degree movie. This task was very difficult as nobody do it in full. Nevertheless, the more good answers were for spherical film than for 2D one. It means that spectators distinguished surrounding environment better when they could turn the world without restraint and see what they wanted to then.

All spectators had problems with telling what type (sequential or simultaneous) of actors' motion was presented in the movies. Cuts and changing shot actions in 2D film bluffed the generality of respondents while persistent shots in spherical movie aided them in recognizing these motions well.

The feel of time is usually disturbed in films and spectators cannot measure it correctly. Major of respondents evaluated it wrong in classical movie while spherical film gave genuine sensation. Lack of shifting shots and sustained actors' motion made participants determine the real length of movie. One part of survey researched how long it took to find some information in the movie. Respondents were looking for vacuum cleaner in the hotel corridor. While they were watching 2D film, they had to wait until they saw it in the action shot. Finding this object was much faster when people could turn the image in 360 degree movie.

The survey participants' opinions about documentation of competition were divided into 3 types. First group was surprised and glad about getting opportunity to feel like contributor of this event. Second one enjoyed this type of film but they preferred if it would be kept as additional form of documentation, not the basic one. The last group thought that turning the movie image was tiring and they did not want to do it all the time. They were open to suggestion of using VR cardboard instead of player on tablet. They were supposed to twirl their head then what is much more natural.

4 Conclusions

360 degree movie is a new phenomenon as there are not hard-and-fast standards like film resolution or projection mode. All of creators are still learning and looking for the best solutions. Spherical movie is very similar to 2D film but some details differ diametrically.

The preproduction process with preparing supportive materials is different. 2D films consist of many shots and needs cut treatments so that camera's field of view can be changed. Spherical movie presents one scene all the time because it is a spectator, not a creator, who decides what is visible in the shot. In this situation, making classical storyboard is meaningless. The postproduction of 360 degree movie also differs from classical film's process. There must be changed projection mode from global to panoramic and covered the invisible pixels with alpha level with special prepared graphic then. Spherical movie allows spectators to recognize surrounding environment correctly as it shows the world in 360 degree. Adjacent objects are located next to each other while in changing shots they are in random order. Opportunity of turning the movie image allows spectators to find particular information faster than waiting until it appears in 2D film. Classical camera's field of view is much smaller than one in spherical camera – some models can even record 360x360 degree environment. This way, the going on action can be shown simultaneously while 2D film must use cuts and changing shots.

360 degree movie is widely known but it is not distinguished in people's awareness. When they start watching spherical film without clear comment about features of this technology, they are not able to use it. To avoid this situation, it is necessary to add some information in the player at the begging of film projection – maybe graphic showing turning the movie image with clicking on the mouse will be sufficient. People receive the documentation of competition positively as they can feel like participants of this event. They were immersed in this environment totally. However, changing the movie image differs opinions from each other. One group thinks that it allows them to discover the movie world on their own account but another group finds it being tiring. There is a chance they would change their mind if cardboard or other VR glasses were used instead of tablet.

Today's new problem is watching this type of movies in a group of people. Individual reception is available on many ways however spherical film projections simultaneously are still undiscovered. However, we must remember that 360 degree movie is still a young, but vigorously developed phenomenon. It can stay with us for a long time if we only find attractive form for every spectator.

References

[1] Woeste, H.: *A History of Panoramic Image Creation*. http://www.graphics.com/article-old/history-panoramic-image-creation, 26.02. 2016.

[2] Kwiatek, K.: 360° *Interactive Storytelling*. University of Glasgow, 2016, http://www.academia.edu/1985746/360-degree_storytelling, 19.02.2016.
[3] Huang, K.-C.; Chien, P.-Y.; Chien, C.-A.; Chang H.-C.; Guo, J.-J.: *A 360-degree Panoramic Video System Design*. National Chiao Tumg University, Taiwan, 2014.
[4] Grau O.: *Virtual Art: From Illusion to Immersion*. Massachusetts Institute of Technology Press, 2004.
[5] Crawford, Ch.: *Chris Crawford on Interactive Storytelling (2nd Edition)*. New Riders; 2 edition (December 27, 2012).
[6] Cavazza, M.; Pizzi, D.: *Narratology for Interactive Storytelling: A Critical Introduction*. TIDSE 2006: Technologies for Interactive Digital Storytelling and Entertainment pp. 72-83.
[7] Charles, F.; Lugrin, J.; Cavazza, M.; Mead, S.J.: *Real-time cinematic camera control for interactive narratives*. ACE '05 Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology, Pages 369-369.
[8] Si, M.; Thue, D.; André, E.; Lester, J.C.; Tanenbaum, J.; Zammitto, V.: *Interactive Storytelling*. Fourth International Conference on Interactive Digital Storytelling, ICIDS 2011, Vancouver, Canada, November 28 – 1 December, 2011. Proceedings.
[9] Spierling, U.; Grasbon, D.; Braun, N.; Iurgel, I.: *Setting the scene: playing digital director in interactive storytelling and creation*. Computers and Graphics, Volume 26, Issue 1, February 2002, pp. 31-44.
[10] Benicewicz-Miazga A.; Klauziński E.; Góra-Klauzińska A.: *Cyfrowa Fotografia Panoramiczna*. pp. 17-33, 2011.

Karolina Nowak Graduate of Faculty of Technical Physics, Information Technology and Applied Mathematics at Lodz University of Technology. Her speciality are Computer Graphics and Multimedia. At present she is a PhD student in the same institute and she studies Virtual Reality with its prospects. She also gets involved in marsian rover project that wins a lot of awards so far (e.g. Rover's World Championship).

Rafał Szrajber Architect, lecturer at the Computer Science Institute at Lodz University of Technology and Strzemiński Academy of Art in Lodz, photographer, new media artist. He teaches New Media in Architecture, Interface design, Computer games design and Photography. The top priority of his workshop is the use of cultural heritage and identity as a carrier of information. Along with Jarosław Andrzejczak creator an idea of information balance in presentation of architecture and works of art via new media. At present he is doing research in the field of virtual reconstruction, the purpose of architecture in computer games and virtual environment representations and creations. Winner of national competitions: architectural, photographic as well as enterprise in science.

Adaptation of the FAtiMA emotion engine for use in computer games

Mateusz Polewaczyk, Marcin Daszuta,
Dominik Szajerman, Adam Wojciechowski

Lodz University of Technology, Łódź, Poland
dominik.szajerman@p.lodz.pl

Abstract

Artificial intelligence (AI) in computer games is very important since the beginning of video game industry. Over the years artificial intelligence is changing, adding new mechanisms to improve overall non-player characters (NPC) behavior. There was many tries to implement flexible and intelligent agents in video games with many different results. Currently AI technologies allows making advanced decision making systems, but ultimately NPCs are mechanically behaving anyway. This paper tries to approach said problem by implementing adaptation of FAtiMA Agent Architecture which adds an emotional aspect to AI. This extend make the architecture more flexible and real-like.

Index Terms

computer games, artificial intelligence, fatima

1 Introduction

For many years decision-making process of non-player characters (NPC) in computer games behave mechanically, which leads to the feeling that the gameplay is static regardless of what is happening around. That is why developers are constantly trying to overcome this barrier in different ways. One possibility that can help in this is adding emotions to agents AI. That will make non-player characters to react more natural to what happens around and then choose best action based on emotional state.

These days, there are many emotional models and each of them has advantages and disadvantages. One of best examples is FAtiMA (**F**earnot **A**ffec**TI**ve **M**ind **A**rchitecture) created by João Dias and Ana Paiva (2005) in which the generation of emotions is inspired by OCC model [1] (named after its creators Ortony, Clore and Collins, 1988). The OCC model is currently treated as a standard model for emotion synthesis. According to the OCC model, every situation that a character may encounter can be evaluated in an appraisal process [2], based on specific criteria and result in multiple emotions of different intensities [3]. Figure 1 shows an overview of the OCC approach. The OCC is commonly used in agent architectures for modeling emotional behaviour [4]. This model has planning capabilities [5], where emotions and personality play a major role in agent's behaviour [6]. There are already several implementations of this model, such as ORIENT [7], FearNot! [8] or Model Of Empathy [10]. ORIENT is a game for teenage users which aims at cultural and emotional learning by engaging adolescents in a role-play with virtual characters in a virtual world. Game was designed to be played by group of 3 users, as input was used WiiMote controller, a dance mat and mobile phones [9]. FearNot! is other application, where empathy is at the center of the interaction between learners and characters. It was developed to tackle and eventually help to reduce bullying problems in schools [8].

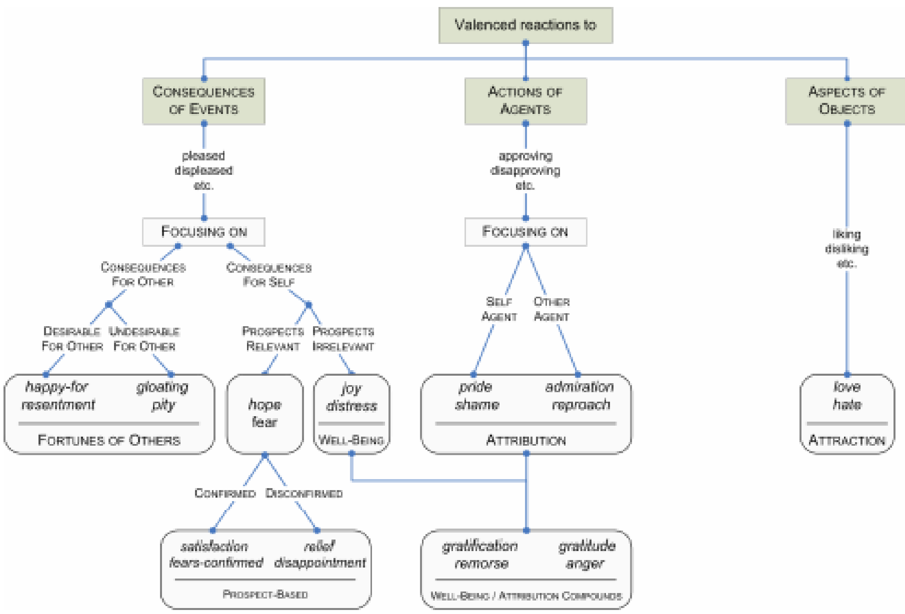


Fig. 1. The OCC model [3]

The aim of this paper is thus to present adaptation of the FAtiMA for use in computer games, and describe how it works. In the paper we give, organization is as follows. In the Section 2, we give a brief overview of the FAtiMA model. In the Section 3, an implementation description of the emotion engine is given and in the Section 4, a sample simulation is described to show opportunities of using this model. Last section shows aggregate results for different agents settings. New approach to emotional agents include simplifiing architecture of FAtiMA to decrease resource consumption with keeping its useful aspects in case of influence on agents actions. In game development it is very important to keep possibly low amount of calculation for every element. High connection with game performance pushed research to focus on shorten adaptation to only needed elements. Effect of such approach is new simplified adaptation of FAtiMA with the same possibilities for certain type of use in video game. Work on FAtiMA adaptation and ways of use result in first tests for implementational correctness and emotional influence on NPC

(Non-Player Character) behaviour. Final outcome of tests are very promising and will need further actions towards examining possibilities of model.

2 FATIMA ARCHITECTURE

The association of the agent emotions with the events is determined by appraisal component. FAtiMA is an extension of a BDI architecture [6] and implements two main behavioural processes: reactive and deliberative [1]. The reactive layer matches events with a set of predefined emotional reaction rules providing a fast mechanism to appraise and react to a certain situation. The deliberative layer generates emotions by looking at the state of current intentions, more concretely whether an intention was achieved or failed, or the likelihood of success or failure [6]. In this architecture emotional status affects agent's motivations, priorities, drives and relationships [11]. Figure 2 shows simplified FAtiMA model. External events correspond to events that happen in the environment, while internal events correspond to events that are triggered by architecture's internal processes [5].

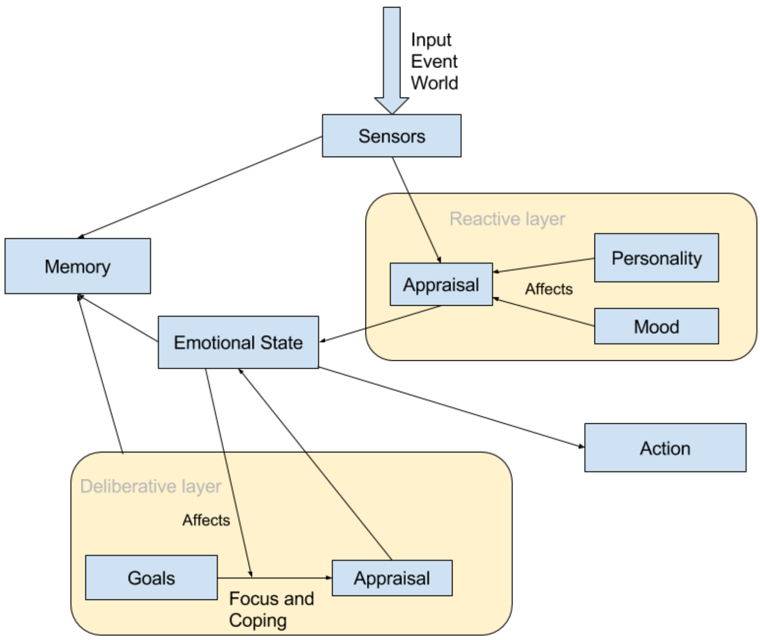


Fig. 2. FAtiMA model

This model is also possible to extend with other layers, such as motivational layer or cultural layer. Motivational component can process basic drives like energy or body integrity. It can take part in deliberative layer to make choice among different objectives. This allows to prioritize needs that agent has. Cultural layer handles rituals, symbols and cultural aspects. Yet, those layers are not necessary for correct behaviour of architecture. This flexible advanatage of FAtiMA is possible due to layered structure of architecture. Every layer except reactive and deliberative can be replaced or not included. As one of the purposes of this work is to create possibly simple component of NPCs AI, additional components are not included in further research [5].

The architecture includes 22 different emotions from OCC model with configurable activation thresholds and decay rates [1] what gives possibility to create large set of different personalities. An emotional threshold specifies a character's resistance towards an emotion type, and the decay rate assess how fast does the emotion decay over time [12]. Decay function proposed by Picard [13] characterizes intensity of emotion as a function of time. At any time (t), the vale for the intensity of an emotion is given by the formula 1:

$$Intensity(t) = Intensity(t_0)e^{-bt} \tag{1}$$

The parameter b determines how fast the intensity of this particular emotion will decrease over time. The value $Intensity(t_0)$ refers to the value of the intensity parameter of the emotion when it was generated. When the value of $Intensity(t)$ reaches a predefined threshold, the emotion will no longer be part of the agent's emotional state [12].

Except emotions, which are temporal, there are other aspects of agents psychology. This engine considers also mood and personality of agent. Mood represents an overall valence of the agent's emotional state and is also used to influence the intensity of emotions. Agent with a good mood has tendency to experience positive emotions and agent with bad mood has tendency to experience negative emotions. Mood decay over time, just like emotions. It continues until it reaches its neutral value with one difference, mood decays linearly.

Our extension considering the personality is shown in the formula (2). It introduces Personality coefficient which is the target value for a given pair of emotions.

$$Intensity(t) = Intensity(t_0)(e^{-bt - \text{Personality}} + \text{Personality}) \tag{2}$$

3 IMPLEMENTATION

During reasearch presented in this paper adaptation of FAtiMA model was implemented and tested. As it was written above we have 22 emotions, they are combined in eleven pairs. Each pair contains contrasting emotions, for example Joy and Distress. The pair is represented by a range from -1 to 1, where negative value corresponds to negative emotion, in this case Distress and positive value corresponds to positive emotion – Joy.

An easy interface has been provided to set up the emotions, which is shown in the Figure 3. This makes it easy to adjust the parameter values to achieve the desired results.

Fig. 3. Emotions configuration

For the purposes of this article, we focus on only one pair of emotions (Joy / Distress) to show the impact of emotions on a simple example.

The implementation of emotional engine was done in Unreal Engine 4 due to the fact that this engine provides many useful things connected with AI such as eg senses. As shown in the model (Fig. 2), emotions are generated on the basis of external or internal events. In this place with the help come the senses (sight, hearing) that are contained in the *Perception Component*, one of the engine components. To use it is needed to assign a function to receive events from it.

```
PerceptionComponent->
OnTargetPerceptionUpdated.AddDynamic(
this, &ATestCharacter::
OnTargetPerceptionUpdated);
```

Upon receipt of the event then it is valued according to the adopted model and based on the generated emotion the appropriate action is taken.

4 Test Case

It was necessary to create test scene with emotion module equipped agents. They would be moving on a loop around the simple scene, while stimuli generating objects were placed around the path.

Thus in a certain area of the game world there are four agents: J, K, O and Z. All of them move along same path, but each one has other configuration options set:

- Z is without emotional engine and it is moving with constant speed, therefore it is a reference point.
- J, K and O have an added FAtiMA engine:
  - O has set neutral personality,
  - J is optimistic,
  - K is pessimistic.

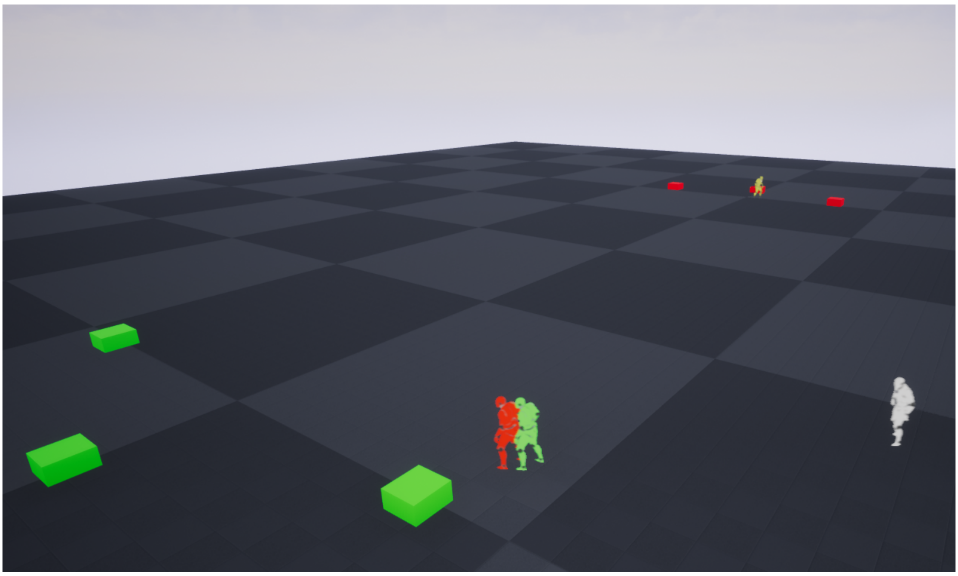


Fig. 4. Agents on Test map

There are six elements on the path that trigger positive and negative emotions alternately. First three positive, then three negative. Based on current emotional state of agent, its movement speed is set. Negative emotional state increases speed whereas positive state decreases it. Based on this test case, you can see how emotions affect agent actions.

The effects of emotions were tested in two different variants: in the first variant emotions were delivered with impulses, while in the second the emotions were delivered continuously in time.

5 Results

Figure 5 shows how agents speed changed in time on the basis of received stimuli for first test variant (impulses). Constant speed of agent without emotional engine was set to 600. Depending on the value of emotion, the agent speed value was set between 300 and 900.

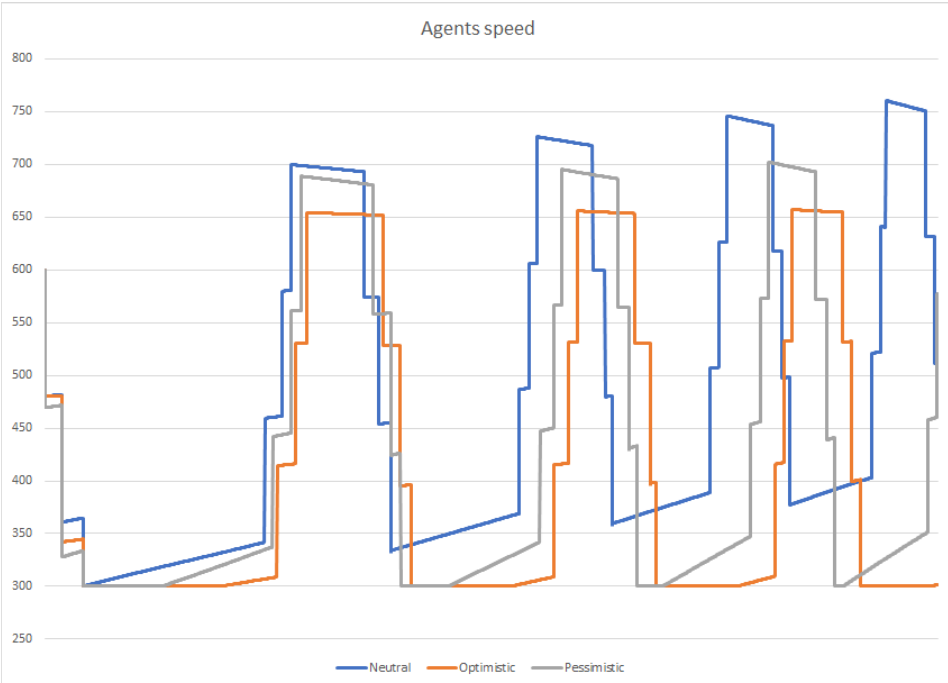


Fig. 5. Agents speed (impulse emotions)

To present possible differentiation in emotional influence on action depending on character proposed were three types: Neutral, Optimistic and Pessimistic. Neutral is most stable emotionally, it means that has lowest vulnerability to outside events. Optimistic has better reactions to positive stimulant and less caring about negative. Finally, Pessimistic is exact opposition to Optimistic. Every peak of speed in the Figure 5 presents reaction on negative stimulant and every trough reaction on positive one. Slight changes of speed towards default speed (600) are connected with decay factor of emotional state that every agent has. Explanation of this change is calming down in areas were no stimulants are visible. Every impulse is marked as point where high velocity shift starts. Result of impulsive stimulant approach is almost discrete change of speed in every case of personality.

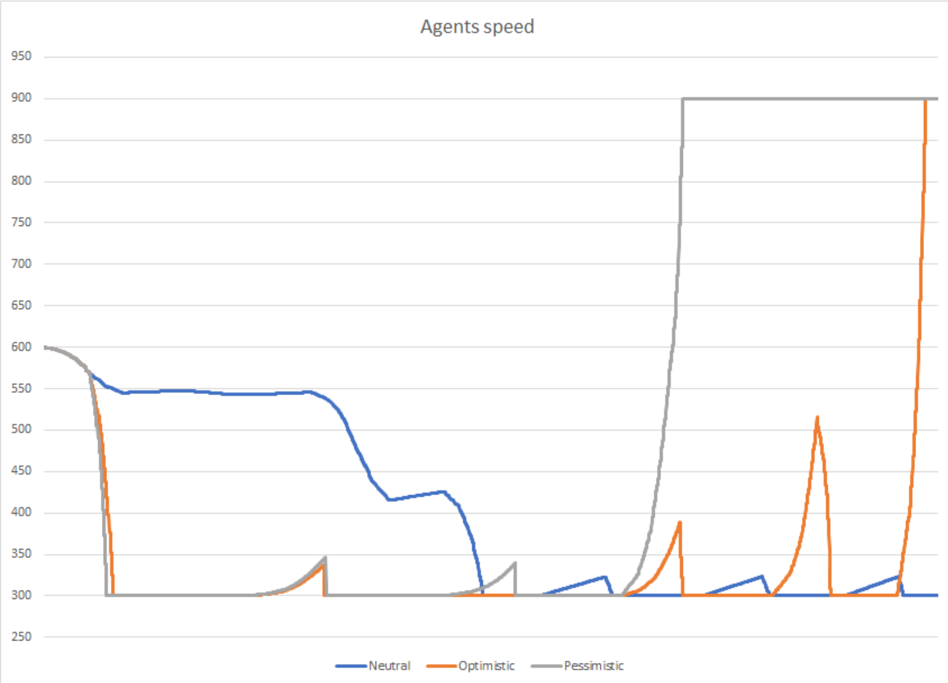


Fig. 6. Agents speed (continuous emotions)

Figure 6 shows the second test variant (continuous emotions stimulation).

In this case instead of impulsive stimulant continuous influence on emotional state was set. In the same test environment behaviour of NPCs is quite different. Opposite types of stimulants have reflection in speed peaks and troughs. Yet, agents actions separate test into two parts. First one at the beginning of diagram presents higher influence of possitive stimulant. Even though in the same time negative were also present. In the second part Negative had higher influence. This change could have connection with irregular speed change what caused different amount of time in range of different stimulants. This final fluctuation resulted with great differences in speed level at the end of test. Such situation should not have place in not looped trajectory of running NPCs.

As can be seen depending on the agents personality set, it behavior is diametrically changing. This translates into a great ability to differentiate agents with just a few parameters.

6 Conclusions

In this paper, the adaptation of the FAtiMA model was presented and discussed. The results attained in the test are promising and give a wide field for improvements. They suggest that the agent's emotional aspect has a big impact on improving the realism and naturalness of their behaviour, which directly translates into gameplay quality in computer games.

In the future, we want to use all eleven pairs of emotions. First is planned to be added emotions love-hate and pride-shame. This will allow us to create more complex scenarios. Another important thing is the actions that are done under the influence of emotions. With more emotion available, we can do more different actions. Therefore, it will be required to make a transparent interface that will easily define the actions and their dependencies.

ACKNOWLEDGMENT

This work was supported by The National Centre for Research and Development within the project "From Robots to Humans: Innovative affective AI system for FPS and TPS games with dynamically regulated psychological aspects of human behaviour" (POIR.01.02.00-00-0133/16).

REFERENCES

[1] Pereira, G.; Brisson, A.; Dias, J.; Carvalho, A.; Dimas, J.; Paiva, A. and others: *Non-Player Characters and Artificial Intelligence*. Psychology, Pedagogy, and Assessment in Serious Games, IGI Global, 2013.

[2] Sánchez, L.P.: *Gaming with Emotions: An Architecture for the Development of Mood-Driven Characters in Video Games*. CETINA, 2013.

[3] Kessler, H.; Festini, A.; Traue, H.C.; Filipic, S.; Weber, M.; Hoffmann, H.: *SIMPLEX – Simulation of Personal Emotion Experience – A new Model of Artificial Emotions*. University of Ulm, Medical Psychology, Institute for Media Computing.

[4] Kiryazov, K.; Grinberg, M.: *Integrating Emotions in the TRIPLE ECA Model*. COST 2102, 122–133, 2010.

[5] Dias, J.; Mascarenhas, S.; Paiva, A.: *FAtiMA Modular: Towards an Agent Architecture with a Generic Appraisal Framework*.

[6] Lim, M.Y.; Dias, J.; Aylett, R.; Paiva, A.: *Creating Adaptive Affective Autonomous NPCs*.

[7] Ruth, A.; Natalie, V.; Elisabeth, A.; Paiva, A.; Enz, S.; Hall, L.: *But that was in another country: agents and intercultural empathy. In Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems – Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 329-336, 2009.

[8] Paiva, A.; Dias, J.; Sobral, D.; Aylett, R.; Woods, S.; Hall, L.; Zoll, C.: *Learning by feeling: Evoking empathy with synthetic characters. Applied Artificial Intelligence*. 235–266, 2005.

[9] Ruth, A.; Nazir, A.; Lim, M.Y.; Rizzo, P.; Enz, S.; Hall, L.: *ORIENT: An Inter-Cultural Role-Play Game*.

[10] Rodrigues, S.; Mascarenhas, S.; Dias, J.; Paiva, A.: *I can feel it too! emergent empathic reactions between synthetic characters. In Affective Computing and Intelligent Interaction and Workshops*. 3rd International Conference IEEE, 1–7, 2009.

[11] Louchart, S.; Aylett, R.: *From Synthetic Characters to Virtual Actors*. AIIDE'07 Proceedings of the Third AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, pp. 88-90, 2007.

[12] Dias J.; Paiva A.: *Feeling and Reasoning: A Computational Model for Emotional Characters*. In: Bento C.; Cardoso A.; Dias G. (eds) Progress in Artificial Intelligence. EPIA 2005. Lecture Notes in Computer Science, vol 3808. Springer, Berlin, Heidelberg, 2005.

[13] Picard, R.: *Affective computing*. MIT Press, Cambridge, 1997.

Mateusz Polewaczyk Student of the last year of computer science at the Technical University of Lodz. For many years his passion is computer games, thats why he is developing in this field. He like to use the UE4 game engine, so far all the games have been created in this engine, including: survival game for engineering thesis, mobile game for the museum contest, VR game for VR Challenge 2016 or just created an emotional engine.

Marcin Daszuta Ms graduate of the second degree studies in Information Technology at the Faculty of Technical Physics, Information Technology and Applied Mathematics at Lodz University of Technology in specialization Computer Simulation and Games Technology and Ph.D. student at the same Faculty. Laureate of ZTGK tornament in 2013 and participant of Imagine Cup competition in 2015. Co-author of a few scientific publications. Participant of scientific research project "From robot to human, innovative technology allowing to fluent and dynamic regulation of influence cognitive and emotional psychology on actions of AI" as developer. Employee in Ericsson Ltd. company as software developer for 2 years. Interested in computer games, computer grathics, virtual reality, computer stimulations, mobile systems, mobile applications, artificial intelligence, level design in computer games.

Dominik Szajerman was born in Łódź, Poland, in 1977. He graduated in computer graphics and obtained the Ph.D. degree from Department of Technical Physics, Information Technology and Applied Mathematics at Technical University of Lodz, Poland, in 2001 and 2010, respectively. During his pre- and postgraduate experience he was involved in a number of research projects in the areas of computer graphics, computer games and human-computer interaction. His academic interests lie in the computer graphics, GPU processing, computer games development and image processing.

Adam Wojciechowski PhD, DSc, Institute of Information Technology, Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology.

Supervisor of an IT teaching specialisation on Computer Games and Simulations. Since 2016 vice-Dean for Development and Business Cooperation at the Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology. Since 2016 vice-chairman of the GameInn Steering Committee in the National Centre for Research and Development. The Lodz University Rector representative in the Lodz Game Dev Cluster and Faculty Dean representative in the ICT Central Poland Cluster.

Research interests encompass: human-computer interaction, cognitive computing, computer graphics, computer vision, computer games, virtual environments, computer simulations and visualizations, computer animations, biometry and machine learning. Author or co-author of over 50 research papers. Coordinator of 3 and participant of 13 national and international research and development projects, financed by Polish and European programs.

EBDI emotion engine adaptation to video game applications

Krzysztof Stengert, Marcin Daszuta,
Dominik Szajerman, Adam Wojciechowski

Lodz University of Technology, Łódź, Poland
dominik.szajerman@p.lodz.pl

Abstract

Video game developers were forced to design and develop artificial intelligence and decision making processes since the beginning of the industry. Algorithms used to create non-player-characters (NPCs) have grown both in terms of quantity, quality and simplicity of implementation. The currently existing solutions are very flexible, precise and easy to understand and use even for non-programmers. However, there is one reoccurring issue with virtual actors in games which is the believability of NPC. This paper tries to solve said issue by implementing EBDI (Emotion-Belief-Desire-Intention) emotion model which combines SIMPLEX based emotion module with BDI agent architecture and gives developers ability to easily define the personality of character which gives believable interaction with virtual actors.

Index Terms

computer games, EBDI, BDI, SIMPLEX, emotion engine, OCC model

1 INTRODUCTION

Since the beginning of the video games industry, the developers faced the problem of creating smart and challenging non-player characters (NPCs) and their artificial intelligence. During many years and thousands of published games few AI systems were designed and developed even to mention the path-finding algorithms, navigation mesh generation algorithms, decision making systems or adapting enemies. At some point developers focused mostly on decision making algorithms which resulted in such solutions as decision trees, state machines and behavior trees. These systems allowed programmers and designers to easily create, manage and extend existing AI behaviors which lead to stunning experiences for gamers. However, there is one reoccurring issue with every algorithm that is based on rational decision making. The issue is believability of developed character – the player could learn how a NPC works and behaves which makes the AI predictable and therefore easy to beat. To avoid this issue developers and researchers did some deeper analysis on how to extend existing AI algorithms and solutions to ensure better experience for the players. This scientific work resulted with new approaches to the issue. One of ideas was to integrate emotions models in currently existing decision-making algorithms. The second developed solution was to create new engine from basis that would allow to introduce emotional models into video games. The most commonly used emotional model is the one proposed by Ortony, Clore and Collins which involve the appraisal process to decide whether an event is positive or negative to NPC and what kind of consequences it should have. The other emotion models that was used in the research are the Scherer's model and many different ideas to enhance existing solutions. In this paper an approach to believability issue is designed and developed by adapting existing models and architectures and combining them to create believable, emotion-based agent for video games industry. Firstly, the few existing solutions are presented and explained. Then, the proposed solution is described along with implementation issues and technical aspects. The next section describes test scenario with obtained results. The last part of this paper gives conclusions and discussion about them with predictions what can be done in future to extend and/or improve proposed solution to make it more usable in video games.

2 Related work

During research the developers managed to develop and adapt few emotions engines to avoid believability issue in video games artificial intelligence. Those engines include emotion generating, memory of a NPCs, emotional state and decision making based on current state of an agent.

2.1 OCC

The most commonly used emotional model is the one proposed by Ortony, Clore and Collins, which involves the appraisal process deciding whether an event is positive or negative for the NPC and what kind of consequences it will have [1], [2]. This process is called appraising and is main part of the model.

The Figure 1 represents the OCC model overview. The consequences of events are analyzed in terms of getting agent closer to accomplish any of its goals. If the agent are getting further from achieving goals then the desirability of the event is low and negative emotions are generated (such as fear or distress). Events that make agent being closer to accomplish some goal have high desirability and result in positive emotions (such as joy or hope). This process can also analyze events that have impact not only on the agent but also on the others – resulting in emotions depending what agent wishes to happen for the others (i.e. if agent wish to hurt his enemy then the event of enemy being hurt is undesirable for the enemy but very desirable for the agent). Appraising actions results in determining the praiseworthiness of the action. Depending on the actions source (if it was the agent himself who performed the action or some other agent) the analyze of the actions can result in one of the emotions pair – pride/shame (the source of an action is agent himself) or admiration/reproach (the source of an action is the other agent). However, the more agent identify with the other agent that performed the actions, the higher chance of analyzing other agent action as actions performed by self (i.e. parents being proud of their child playing in a winning football team). The last appraisal category is the analyzing aspects of objects – the agent can either like or dislike the object resulting in love/hate emotions pair [2].

2.2 SIMPLEX

SIMPLEX (Simulation of Personal Emotion Experience) [2] is an example of mentioned emotion engines. The model is consisting of four main modules: the appraisal process (with input events and agent's goals), the personality of the NPC, the mood-state and the memory of developed agent (Fig. 2).

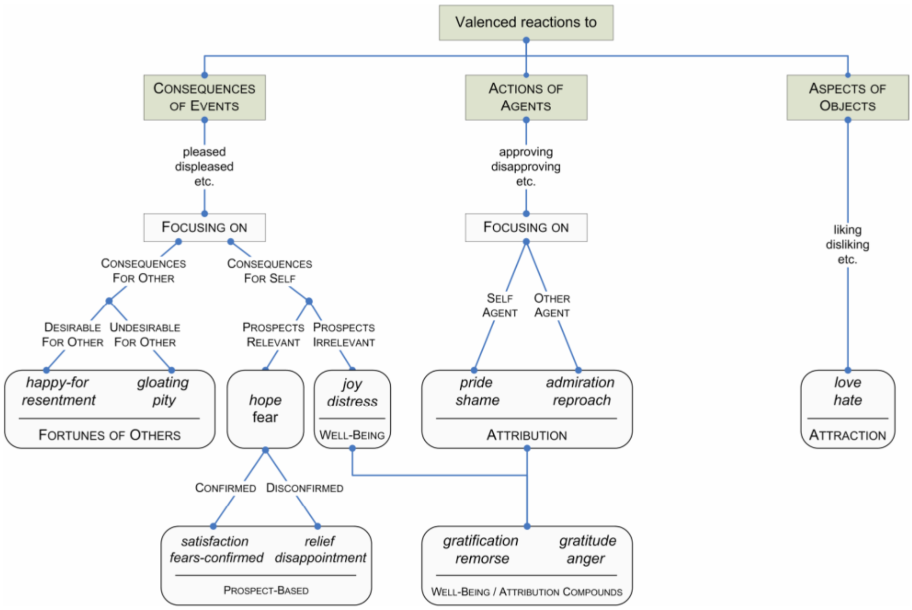


Fig. 1. The OCC model (source: [2])

The appraisal process uses OCC (Ortony, Clore and Collins, 1988) emotion model to generate emotions based on input events and current agent's goals. The event is examined in terms of desirability of its consequences and the praiseworthiness of agent's actions [2]. This step involves personality of NPC to influence determining whether an event is positive (desirable) or negative (undesirable) for character along with deciding if the possible actions will result in generating more desirable events.

The personality module is based on Five Factor Model (FFM, a.k.a. OCEAN, Big Five Personality Traits) introduced by psychologists McCrae & Costa [4]. This model assumes that the personality of a character can be described using five traits: extroversion, conscientiousness, agreeableness, neuroticism and openness to experience. The extroversion determines how the positive emotions are experienced. High scores in that trait define extraverted person who enjoys every positive emotion and low extroversion means introverted person who likes loneliness. The conscientiousness is used to model people with high emotional stability – the more person is conscientious the more it plan a lot, think about everything and is more stable. The agreeableness is about how much does a person care about other characters. The lower agreeableness the more the character is selfish and puts its own good above any other. The agreeableness also determines the tendency to care about the actions consequences for others. The neuroticism can be explained as nearly an opposite for extroversion. It is a tendency to experience negative emotions and exaggerate them. However, a high neuroticism also means that the person is more sensitive and reacts

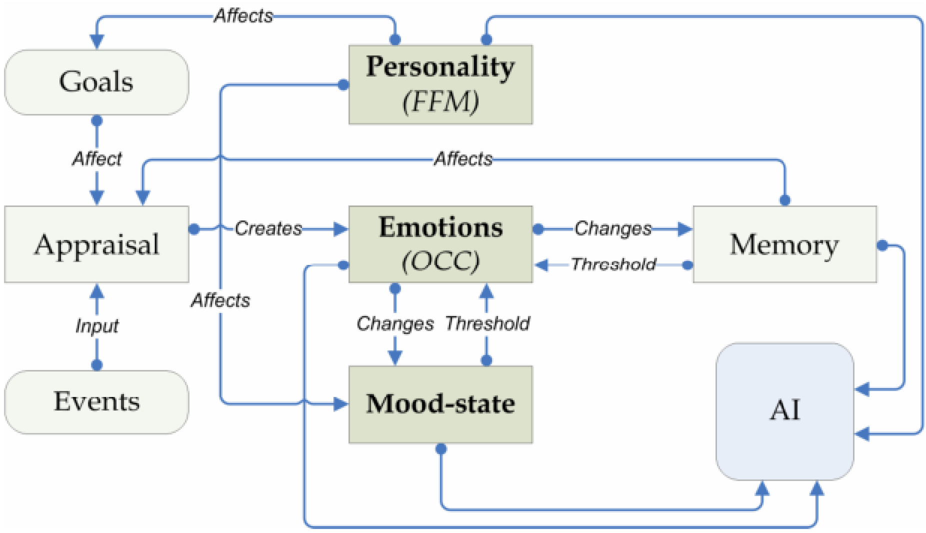


Fig. 2. SIMPLEX model (source: [2])

emotionally to meaningless events. The openness to experience defines creativity and curiosity of a person. High values in this field defines people who are not emotional stable and quickly forgets about the consequences of past events in the search of new challenges [2].

The mood-state module represents current emotional state of an agent using three dimensional PAD (Pleasure, Arousal, Dominance) space [2] introduced in [3]. This state can be influenced by generated emotions (also represented in PAD space) from appraisal process – they changes it by adding their values to the mood-state and resulting in a new state. There is also a decay applied to the mood-state which means that the character after experiencing an emotion slowly returns to base emotional state.

The last main module in SIMPLEX model is the memory. It is simple structure holding pairs of events and emotions generated by that event. The model uses memory to influence emotion generating – if an event occurred in the past then the agent can more precisely determine its desirability by having knowledge about emotions generated previously.

2.3 FAtiMA Modular

Another emotion engine is the FAtiMA (**F**earnot **A**ffec**ti**ve **M**ind **A**rchitecture) is an architecture of agent in which the emotions and personality have a high influence on agent's behavior and planning processes [6]. This architecture was successfully used in several scenarios such as FearNot! [10] and ORIENT [9]. The modular version of this architecture [6] allows to use only unnecessary components or comparing appraising models by changing appraisal component (OCC, Scherer). The FAtiMA Modular model is made of few layers: the core layer (called FAtiMA Core), the

Reactive Component, the Deliberative Component, the OCCAffectDerivationComponent, the Motivational Component, the Theory of Mind Component and the Cultural Components. It is not obligate to use them all at once – the developer can implement only a small subset of this components – the authors ensures that using the core component and one or two additional will result in working emotion based agent (the agent with only core component will not do anything at all).

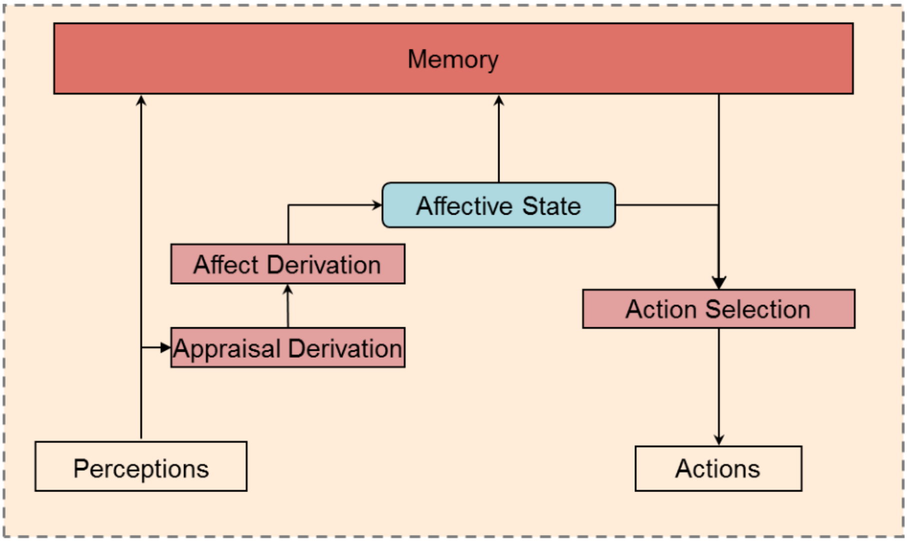


Fig. 3. The FAtiMA Core Architecture (based on [6])

The core component (FAtiMA Core) is presented in the Figure 3. It consists of appraisal derivation, affect derivation, affective state, action selection and memory modules. The appraisal derivation can use either OCC emotion theory, Scherer's theory or any other appraisal process by implementing proper component (the [6] uses OCC model). The agent after receiving perceptions from the virtual environment will pass them to the appraisal derivation as well as to the memory module. The appraisal derivation takes the perception as input and determines the set of variables called appraisal variables (for the OCC model the set will consist of desirability and praiseworthiness). Then the affect derivation has place. It takes the set of appraisal variables as input and examines them to generate the affective states (i.e. emotions or mood). The affective state is simply just the current emotional state of an agent – the emotions or the mood state. The action selection is the place where behavior component is implemented. The core will ask the behavior component whether the action should be performed and what type of action it will be. This module results in action or set of actions that the agent should perform in his current state.

The appraisal derivation and the affect derivation both serve as a appraisal process. The reason of dividing this appraisal to two distinct modules comes from the Structural Appraisal Theories [11] and the Set based Formalism [12] which identify three main processes in Appraisal Theory: perception, appraisal and mediation.

As mentioned earlier the FAtiMA Modular architecture described in [6] is made of several components (the core component is described above):

- Reactive Component – has predefined emotional reaction rules and determines appraisal variables for OCC model based on that rules. The variables are: desirability, desirability for other, praiseworthiness and like. Once the event is perceived the component matches it against a set of rules. Then the values of the variables are determined and passed further.
- Deliberative Component – the second appraisal component which handles goal-based behavior and extends agent about planning capabilities.
- OCCAffectDerivation Component – handles appraisal variables generated by first two components according to OCC emotions theory. This state results in emotions like joy, distress, hope, fear etc.
- Motivational Component – responsible for modeling basic human drives. Additionally it is used to determine the desirability of an event's affect on that drives – if the event lowers one of the need then it is evaluated as desirable and if it raises the need then undesirability of that event occurs.
- Theory of Mind Component – component which serves for creating a state of other agents and simulating their own appraisal processes to determine the desirability for others.
- Cultural Component – adds cultural-dependent behavior of agent based on defined culture by rituals, symbols and cultural dimensions. Uses cultural values to determine the praiseworthiness of an action.

2.4 EBDI

The EBDI architecture proposed in [8] extends traditional BDI (Belief Desire Intention) agent architecture about Emotion resulting in Emotion Belief Desire Intention architecture. The Belief part can be described as agent's knowledge about the environment or more likely as what agent considers as true or false. According to [8] the beliefs can be changed through perception, contemplation or communication. Beliefs from perception are generated from the environment through agent's senses – what the agent sees, what he hears or touches. The communication channel gives beliefs from communication with other agents in the environment. For example one agent can tell another that the enemy is behind the wall. As a result of that message the agent (leaving the reliability of the other) will believe that the enemy is truly behind the pointed wall. Beliefs from contemplation are generated by taking current emotion status and intentions, and revising them. All three beliefs acquiring channels result in beliefs candidates which are updates the beliefs set after the contemplation finishes [8]. The Desire part of EBDI architecture depicts what agent wishes to do based on its beliefs and the agent's state. For example if agent belief that the medicine located somewhere will heal him and the agent was hurt recently then he will desire to walk to the medicine location and take it. The Intention is a desire filtered by agents possibilities which also come from the beliefs. Continuing previous example, if the agent desires to walk for a medicine but there is an enemy on the path then the desire of walking there will be postponed and do not become the intention. The Emotion part serves as additional filter for desires and intentions generation. There are two types of emotions taking into account – the primary emotions and the secondary emotions. The primary emotions come from the perception and communication and are the first emotions that the agent feel when new events occurs. The secondary emotions appear after primary emotions and may not be caused directly from environment but from agent's contemplation process (thinking about primary emotions, for example anger can turn into dislike or hate). The execution of the process of decision making in EBDI architecture can be enumerated as follows:

1) The new information about the environment is acquired through perception or communication resulting in belief candidates.
2) The first feelings about new informations appear – the primary emotions are generated.
3) Re-evaluation of the beliefs happen based on new emotion status and informations obtained in 1.
4) Desires are generated.
5) The intention is generated under the influence of beliefs, desires and emotions.
6) The secondary emotions are generated from the contemplation.
7) If there is a time for deeper consideration (or the emotion status has not changed) then the beliefs, desires and intentions are updated with the influence of secondary emotions.
8) The set of actions (plan) is constructed and executed based on intentions and set of possible actions.

2.5 EBDI with Theory of Emotion Regulation

The EBDI described in 2.4 is not the only attempt to extending BDI agent architecture. Another solution based on BDI is EBDI with Theory of Emotion Regulation presented in [7]. In this model most of the time the agent makes his decision with rational thinking in pure BDI way. However, the emotions have impact on agent's behavior when the actual level of an emotion is beyond the "neutral" emotion level scope.

The model defines two ERL (Emotional Response Level) values for each emotion – one for actual level of an emotion and the second one for the base level of an emotion. The ERL value is in range from 0 to 2, where 0 is the lowest possible value and 2 is the highest. The agent is always going to keep his actual ERL value near the base value. Moreover the model defines input events as a set of parameters (values in range from 0 to 1) defining the impact of the event on the emotions. For example, when agent first time sees an enemy then the event might have 0.9 impact on fear, 0.2 impact on excitement and 0 impact for all other emotions. There is also a one additional variable defined which is the speed of returning from actual ERL to base ERL value and denoted as β. Equation 1 represents the determining of current ERL.

$$ERL_{current} = ERL_{actual} + (ERL_{base} - ERL_{actual}) * \beta \qquad (1)$$

Additionally, the model assumes that every agent has some basic knowledge about other agent internal state and takes actions that will be desirable for others by simulating their internal state changes which will be the result of agent's actions. This is called Theory of Mind (ToM) and present also in previously mentioned FAtiMA Modular (2.3) model. In [7] there is also another theory called Theory of Emotion Regulation which assumes that an agent can regulate emotions that other agents experience by leading them to specific places or showing them specific objects or events. Of course, this assumes also that the other agents will listen to the "supervisor" and will follow him. The emotion regulation is successfully achieved by keeping an ERL value of other agents by "supervisor" and trying to control that value in such a way that it is close enough to some assumed ERL value. For example a team leader should try to keep the team members ERL value for joy as high as possible.

The main differences between this architecture and EBDI architecture from 2.4 section are the presence of Theory of Mind and the way the emotions are handled. In previous system the emotions had great impact on each part of the BDI system but in this model emotions are used to decide whether agent should react in normal, rational way using traditional decision making process or the reaction of the agent should be emotionally and the NPC should abort rational thinking. The Figure 4 shows steps in the EBDI architecture algorithm. First the actual ERL is adjusted based on base ERL value and regulation speed β. Then the ERL value is compared with minimum and maximum values of the ERL. If the current value is in (min, max) range then rational decision making is performed and action is chosen. But if the actual ERL value is beyond the scope of minimum and maximum then rational thinking is skipped and the agent chooses action based on its actual emotional state.

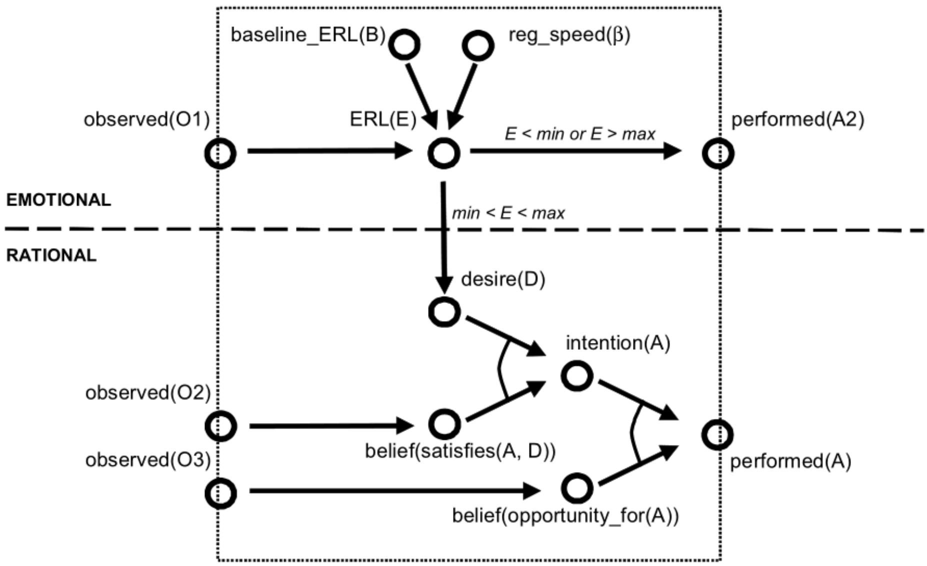


Fig. 4. The EBDI architecture (based on [7])

3 Proposed solution

In this section a solution approaching the believability issue in non-player characters controlled by traditional artificial intelligence algorithms will be described. The proposed model is a combination of SIMPLEX architecture explained in 2.2 based on OCC emotions theory (2.1) with traditional BDI agent architecture which will result in an emotion engine similar to EBDI system described in 2.5.

The presented engine consist of three main modules: the personality of an agent, the emotional state and the BDI architecture. The system is depicted in the Figure 5.

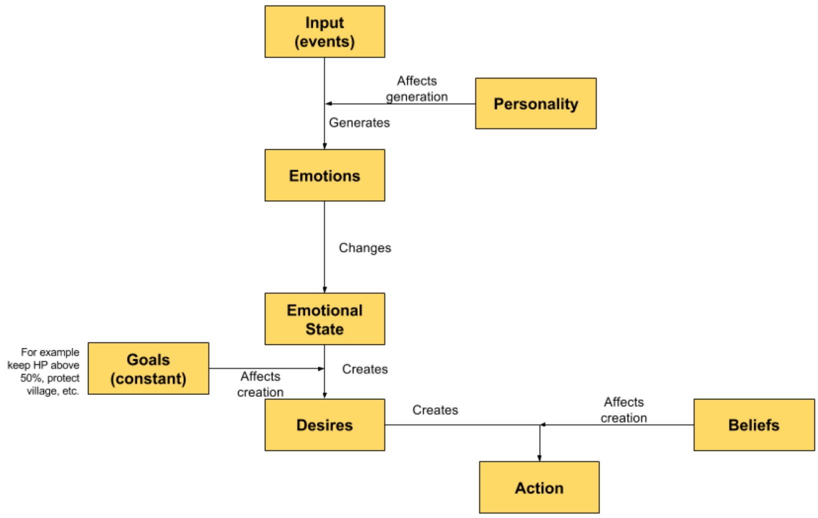


Fig. 5. Proposed EBDI emotion engine

The personality module describes the personality of an agent using Five Factor Model (FFM) thus giving a possibility to design different characters by maintaining just five parameters. All the traits values are in range from -1 to 1. The engine uses personality to influence the emotions generation and to determine base emotional state of an agent. The personality traits are:

- Openness to experience,
- Conscientiousness,
- Extroversion,
- Agreeableness,
- Neuroticism.

The emotional state is a PAD (Pleasure, Arousal and Dominance) space representation of current agent state. It is changed by emotions generated from input events and used for creation of agent's desires in BDI part of the engine.

The BDI architecture used in proposed solution is truncated in such a way that desires are not generated through beliefs. Instead the engine creates desires based only on agent's goals (for example keeping HP (Health Points of the agent) level above some specified value) and the current emotional state of the character. Desires turns into intentions by the influence of agent's beliefs as in traditional BDI architecture.

Emotions are generated based on OCC theory emotions and the generation process is affected by the personality of NPC using three of five traits: neuroticism or extroversion, openness to experience and conscientiousness. The choice between neuroticism and extroversion is made based on whether the event is positive or negative for the agent. All of the three personality traits influence the generated emotion with a different power thus there are traits weights for each of them. The weights take values from 0 to 1 and what is important, the weight for neuroticism has the same value as the weight for extroversion. The equations 2, 3 and 4 give an overview how the emotion is affected by personality.

$$TW = OW + NEW + CW \tag{2}$$

$$I = \frac{O * OW + N * NEW + C * CW + TW}{2 * TW} \tag{3}$$

$$I = \frac{O * OW + E * NEW + C * CW + TW}{2 * TW} \tag{4}$$

where I is the total influence calculated, TW is total weight, OW is openness to experience weight, NEW is neuroticism/extroversion weight (since the weight for this two traits has to be the same), CW is conscientiousness weight, O is openness, N is neuroticism, C is conscientiousness and E is extroversion. Just to remind, weight values (OW, NEW and CW) are in range from 0 to 1 and personality traits (O, N, C and E) values are in range from -1 to 1.

The emotion's presentation used in the engine is the PAD space. As mentioned earlier the personality, defined using FFM, is used also to determine the base emotional state of the character. Thanks to Mehrabian [5], the conversion is possible and uses following equations):

$$\begin{aligned} P &= 0.21 * E + 0.59 * A - 0.19 * N \\ A &= 0.15 * O + 0.3 * A + 0.57 * N \\ D &= 0.25 * O + 0.17 * C + 0.6 * E - 0.32 * A \end{aligned} \tag{5}$$

One must note that in Mehrabian's work he uses sophistication instead of neuroticism but the neuroticism is just simple inverted sophistication (the signs in equations are inverted where needed also).

During the simulation or game the agent need to make some decision. In presented engine it is done by presence of BDI architecture. The agent takes his current emotional state represented in PAD space and based on it and current goals he creates the desire. For example if the current goal is to keep HP level above some specified value but the actual HP level is below that value (resulting in fear emotion for example) then the NPC will try to heal himself in one way or another. So the desire created will be "find a medicine" for example. The agent takes then his beliefs and creates an action that he will perform. So, in presented example, he believe that he saw medicine in one room and combination of "find medicine" desire and "medicine is in X room" belief he will create an action of "go to X room and take the medicine".

Taking short-term emotion nature into account the current emotional state of the agent also changes towards base emotional status calculated at the beginning of the simulation because of the emotion decay. The decay comes directly from the personality of the NPC (conscientiousness, openness to experience and neuroticism) and can be calculated as follows:

$$D = \frac{(O * OW + N * NW - C * CW + MD)}{2 * MD} \tag{6}$$

where O, N and C are openness, neuroticism and conscientiousness, OW, NW and CW are openness weight, neuroticism weight and conscientiousness weight and MD is max decay value (sum of all three weights). The decay is applied every frame to current emotional state resulting in new emotional state. The equation 7 presents the decay effect.

$$ES_c = ES_c + (ES_b - ES_c) * D * deltaTime \tag{7}$$

When it comes to the decision making the engine calculates the emotions pairs values as the distance of current emotional state to emotion. The example calculation for the joy/distress pair is presented by equation 8.

$$JoyDistress = dist(ES_c, Distress) - dist(ES_c, Joy) \tag{8}$$

The $Distress$ and Joy symbols are representation of distress and joy emotions in PAD space. The $dist$ function is the distance in a PAD space which is calculated using Euclidean metric. The $JoyDistress$ is the value of joy/distress emotion pair. The value of any emotions pair is in range from -1 to 1. The obtained values (for each emotions pair) are then used to create the agent's desires. For example when agent's joy/distress emotion pair is closer to 1 (the distance from distress is near 1 and distance from joy is near 0) then the agent can create desire to make gesture of pure joy (victory gesture or similar).

4 TEST SCENARIO AND RESULTS

To test the EBDI emotion engine proposed in 3 the test scenario has been designed and created using Unreal Engine 4 by Epic Games. In the test map there are emotion stimulus in form of "good" and "bad" bonfires. The "good" ones send positive event to encountered agent and the "bad" ones results in negative event. The bonfires send the events all the time the agent is in their range (simple sphere with the center located in the bonfire location). There are also three agents placed on the test map – one without emotion engine (the yellow one) and two with the engine present. Those two agents are differentiate by the personality – one is emotional stable (the green one) and the other one is not stable at all (the red one). The agents runs on the path specified using the spline component. In the scenario there is only one emotions pair chosen and it is the joy/distress pair. As a result the "good" bonfires should generate the joy emotion and the "bad" bonfires ought to generate distress emotion. The decision making process has to control only one variable – the speed of the NPC in specified way. When the agent gets the positive events then he should slow down to be as long as possible in the emotion stimulus range and the negative events results in accelerating to run out of the bonfire range as quick as possible. The agent without emotion engine serves as a reference by running all the time at constant speed which helps to see the difference between the agents – the distance between them changes at time. The emotional state of the agents is represented using colored sphere above their heads where the green color means that the agent's emotional state is very close the joy emotion in the PAD space, yellow color means that the distance from joy equals the distance from the distress and the red color which means that the agent is very close to the distress emotion. The Figures 6, 7 and 8 show the test scenario in action.

The results obtained during the simulation are satisfactory and promising. The agents with the emotion engine behaved in different ways as expected – the emotional stable agent (the green one) was much slower than two other agents since he slowly changes emotional state due to his stable nature and the red one (not stable agent)

Fig. 6. Red agent enjoys (its sphere is green) the “good” bonfires

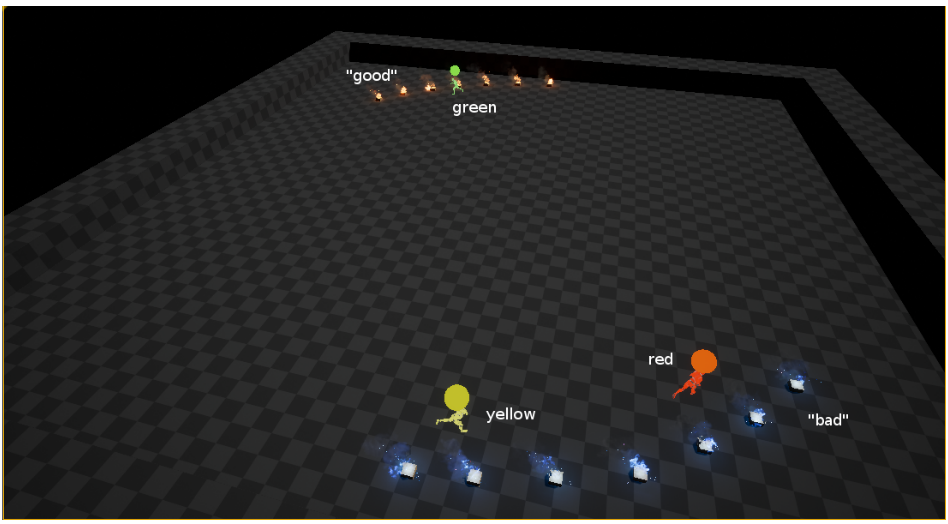


Fig. 7. Red agent is distressed (its sphere is red) by “bad” bonfires

accelerates and decelerates dynamically which also was expected since his emotional state changes fast due to his emotional instability. After a while the red agent doubles the green one – it is also expected because the stable agent does not care much about the negative events (because to achieve the high emotional stability the neuroticism value was set below 0 to ensure that the decay is at low level, see equation 7).

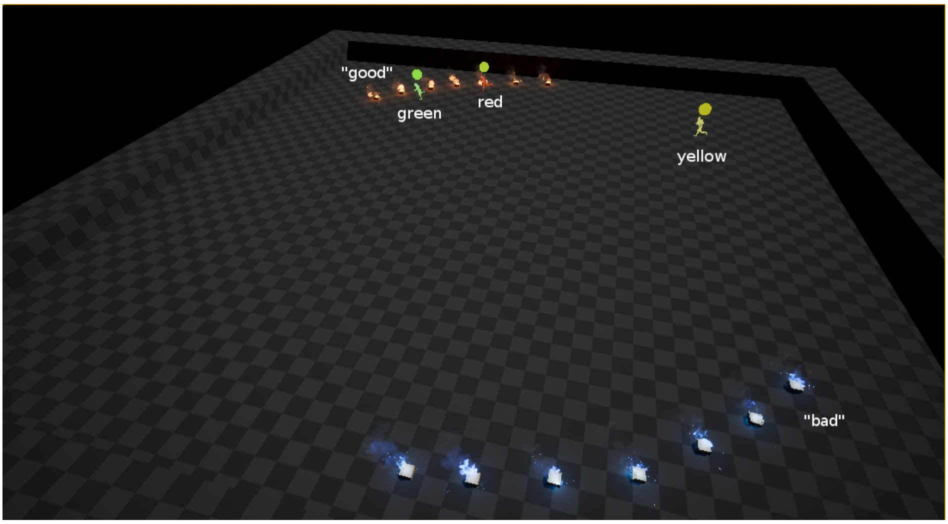


Fig. 8. Red agent will double the green one in a moment

5 Conclusions

This paper tries to approach the believability of the traditional artificial intelligence algorithm and NPCs in video games by proposing an EBDI emotion engine combining SIMPLEX model with BDI agent architecture. The results obtained in the test scenario are promising and give a wide field for improvements. The agents in that scenario behave as expected – the emotion stable agent slowly changes his emotional state and therefore runs at low speed for most of the time. The not stable agent serves as an opposition to the first agent. His emotional state changes all the time dynamically with high rate resulting in frequent, noticeable changes in his speed. The influence of particular personality traits also worked as expected – the agent with high emotional stability with higher extroversion and lower neuroticism experienced the positive emotion stimulus more sensitive than the negatives ones and the second agent with the neuroticism and extroversion reversed comparing to the first one was more sensitive to the negative bonfires.

5.1 Future work

Despite of the promising results and conclusions the authors of this article will continue the research and work on the emotion engine. The first, obvious field in which the engine can be extended are other emotions combined in pairs. The authors plan to extend the count of the emotions pairs from one to eleven including love/hate, happyForOthers/pity, pride/shame and hope/fear. By extending the engine about mentioned emotions pair the development of more complex scenario will become possible.

Adding more than just one variable controlled by the emotion engine is the next obvious step in extending proposed solution. It may be the whole navigation and path-finding process of the NPC as well as gesture performing or recognizing (gestures of other agents) and crowd simulating.

Another field in which the engine might be extended is introducing memory module to the EBDI. Maintaining memory is not trivial problem but may result in additional possibilities of developed solution. The memory will serve as a container in which the emotion stimulus source, the event type and the emotion generated will be stored and used for future emotion generation as well as for the desires creation.

The decision making process is the next field of extending the solution. It may be that adding another algorithm to the process will result in much more believable and easy to develop solution. The authors plan to use behavior trees built in Unreal Engine 4 for the decision making process and use the BDI architecture for setting the current state of the agent (aggressive, patrol etc.).

Acknowledgment

This work was supported by The National Centre for Research and Development within the project "From Robots to Humans: Innovative affective AI system for FPS and TPS games with dynamically regulated psychological aspects of human behaviour" (POIR.01.02.00-00-0133/16).

References

[1] Ortony, A.; Clore, G.; Collins, A.: *The cognitive structure of emotions*. Cambridge University Press, Cambridge, MA, 1988.

[2] Kessler, H.; Festini, A.; Traue, H.C.; Filipic, S.; Weber, M.; Hoffmann, H.: *SIMPLEX – Simulation of Personal Emotion Experience – A new Model of Artificial Emotions*. ISBN: 978-3-902613-23-3, University of Ulm, Medical Psychology, Institute for Media Computing, Germany (2008).

[3] Russel, J.A.: *Evidence of convergent validity on the dimensions of affect*. Paper presented at the meeting of the Psycho-metric Society, Iowa City, Iowa, 1975. Journal of Personality and Social Psychology, 36, 1152-1168.

[4] McCrae, R.R.; Costa Jr.; P.T.: *Validation of the five-factor model of personality across instruments and observers*. J Pers Soc Psychol, Journal of Personality and Social Psychology1987, Vol. 52, No. 1,81-90.

[5] Mehrabian, A.: *Analysis of the Big-five Personality Factors in Terms of the PAD Temperament Model*. Univeristy of California, Los Angeles, August 1996.

[6] Dias, J.; Mascarenhas, S.; Paiva, A.: *FAtiMA Modular: Towards an Agent Architecture with a Generic Appraisal Framework*. Technical University of Lisbon, Lecture Notes in Computer Science, vol 8750. Springer, Cham, 2014.

[7] Bosse, T.; de Lange, F.P.J.: *Development of Virtual Agents with a Theory of Emotion Regulation*. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2008, 461-468.

[8] Jiang, H.; Vidal J.M.; Huhns, M.N.: *EBDI: An Architecture for Emotional Agents*. University of South Carolina, Columbia, SC 29208, 2007.

[9] Nazir, A.; Lim, M.Y.; Kriegel, M.; Aylett, R.; Cawsey, A.; Enz, S.; Rizzo, P.; Hall, L.: *ORIENT: An Inter-Cultural Role-Play Game*.

[10] Dias, J.; Paiva, A.: *Feeling and Reasoning: a Computational Model for Emotional Characters*. In: Bento C.; Cardoso A.; Dias G. (eds) Progress in Artificial Intelligence. EPIA 2005. Lecture Notes in Computer Science, vol 3808. Springer, Berlin, Heidelberg, 2005.

[11] Reisenzein, R.: *Appraisal Processes in Emotion. Theory, Methods, Research, chapter Appraisal Processes Conceptualized from a Schema-Theoretic Perspective: Contributions to a Process Analysis of Emotions*. Oxford University. 2001.

[12] Broekens, J.; DeGroot, D.: *Scalable and flexible appraisal models for virtual agents*. 5th Game-On International Conference: Computer Games: Artificial Intelligence, Design and Education, 2004.

Krzysztof Stengert BSc – a graduate of the first degree studies in Information Technology at the Faculty of Technical Physics, Information Technology and Applied Mathematics at Lodz University of Technology in specialization Computer Simulation and Games Technology – thesis topic: “Behavior trees in computer game engine”. Double fellow of Lodz University of Technology rector’s scholarship. Participant of EU project “Informatyka – kierunek zamawiany na Politechnice Łódzkiej” in the “Computer Graphics in Entertainment and Computer Games” module. Co-creator of published Android game “Donut Madness” developed within EU project. Participant of several game jams including Slavic Game Jam 2015 organized by “KNTG Polygon”, Komiks Game Jam 2015 and Łódzki Game Jam 2015 and 2016. Co-developer of titles published by Teyon – “Bird Mania Party for WiiU” on Nintendo WiiU console, “Crayola Bubbles” on Android and PC using Intel RealSense, “50-in-1 Games Adventure Planet” on Android. Co-developer of Android game “Pangrollin”’ published by Vescom Studios.

Marcin Daszuta Ms graduate of the second degree studies in Information Technology at the Faculty of Technical Physics, Information Technology and Applied Mathematics at Lodz University of Technology in specialization Computer Simulation and Games Technology and Ph.D. student at the same Faculty. Laureate of ZTGK tornament in 2013 and participant of Imagine Cup competition in 2015. Co-author of a few scientific publications. Participant of scientific research project “From robot to human, innovative technology allowing to fluent and dynamic regulation of influence cognitive and emotional psychology on actions of AI” as developer. Employee in Ericsson Ltd. company as software developer for 2 years. Interested in computer games, computer grathics, virtual reality, computer stimulations, mobile systems, mobile applications, artificial intelligence, level design in computer games.

Dominik Szajerman was born in Łódź, Poland, in 1977. He graduated in computer graphics and obtained the Ph.D. degree from Department of Technical Physics, Information Technology and Applied Mathematics at Technical University of Lodz, Poland, in 2001 and 2010, respectively. During his pre- and postgraduate experience he was involved in a number of research projects in the areas of computer graphics, computer games and human-computer interaction. His academic interests lie in the computer graphics, GPU processing, computer games development and image processing.

Adam Wojciechowski PhD, DSc, Institute of Information Technology, Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology.

Supervisor of an IT teaching specialisation on Computer Games and Simulations. Since 2016 vice-Dean for Development and Business Cooperation at the Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology. Since 2016 vice-chairman of the GameInn Steering Committee in the National Centre for Research and Development. The Lodz University Rector representative in the Lodz Game Dev Cluster and Faculty Dean representative in the ICT Central Poland Cluster.

Research interests encompass: human-computer interaction, cognitive computing, computer graphics, computer vision, computer games, virtual environments, computer simulations and visualizations, computer animations, biometry and machine learning. Author or co-author of over 50 research papers. Coordinator of 3 and participant of 13 national and international research and development projects, financed by Polish and European programs.

Adaptation of PSI emotion engine for use in video games

Patryk Kochański, Marcin Daszuta,
Dominik Szajerman, Adam Wojciechowski

Lodz University of Technology, Łódź, Poland
dominik.szajerman@p.lodz.pl

Abstract

There was many tries to implement flexible and intelligent agents in video games with many different results. Now existing AI technologies allow for creating non-player-characters (NPC) in games with advanced decision making and precise problem solving. The one reoccurring issue for virtual actors in games is believability and creating real-like behavior. This paper tries to approach said problem by adapting a simplification of Psi theory and implementing it in suitable way for games non-player-characters, also giving chance to adapt actors personalities. Moreover, first tests were made, based on emotions dynamics implemented in this adaptation. Agents running over the certain trajectory could react in different way to stimulus placed over the testing area. Based on research contained in this paper emotion engines as one described have great potential in creating more human-like behavior of agents in video games.

Index Terms

computer games, games artificial intelligence, Psi theory, artificial emotions

1 Introduction

At the beginning computers were used, to help with complicated mathematical calculations. By using machines one could save many hours of work. Rapid growth of the technology caused the computers to became more popular and commonly available. All new approaches of using PC, led to creation of information technology as a branch of science. During development of IT it became clear that computers have great future in entertainment branch. Many people use their devices to surf the Internet, watch videos, listen to music or playing video games. In this paper we will focus on computer games branch which is undeniably one of biggest and fastest developing part of electronical entertainment. In course of time, gaming migrated from computers to stationary and handheld consoles, phones and other devices capable of running games. In fact, mobile gaming is the fastest developing branch of virtual entertainment.

2 Brief summary of computer games history

First computer games had simple logic and in perspective of new games they were very simple programs. At the beginning those were just text games. Player could pick one of few displayed messages and in case of his/her choice, there were other messages displayed. More advanced games used text symbols (ASCII) to present concept of the world, where game took place. Next step of games evolution was 2D pictures usage instead of ASCII symbols. Creators now could convey their vision of appearance characters and other objects in game environment. In course of time and computers development games could become more complex. Players could experience sounds, nice looking images and more complicated gameplay. Good example of growing complexity could be the use of artificial intelligence to make games more challenging. Next very important step in gaming development was the invention of 3D models. Contemporary games cannot be compared with old simple programs. Nowadays games are filled with complex mechanics providing players entertainment for many hours.

3 Important part of computer games

One of those complex mechanics is artificial intelligence. At the beginning computer games had automatically moving elements of environment regarded as very complex and challenging. Programmers achieved that by simple if-statements and every player was totally satisfied. With growth of game development, there were needs for more complicated artificial intelligence algorithms and decision making mechanics. Because of that, gamers wanted to see more realistic non-playable characters (NPC). Nowadays uncountable number of artificial intelligence algorithms exist. Commonly used algorithms in game development are state machines and behavior trees. These algorithms are huge improvement comparing them with simple if-statements used at the beginning of games production. As the example of the artificial intelligence in computer games non-playable characters can be considered as example. Name of those characters simply explains that their behavior is independent from player actions and they are controlled by predefined algorithms. The reason of using more advanced AI techniques is players awareness. They expect NPC's act like they would behave. In contemporary games the border between real world and virtual one is already blurred. Games where NPCs act unnatural are low rated. That is undesirable state, because people will not spend their money on low quality product. Enemy with low hit points, rushing to player position, ignoring risk of his life can be considered as an example of such situation. Another example more annoying for players can be virtual teammate, who prefers to explore game environment, instead of helping player during any kind of trouble.

4 Idea of improvement

Game developers focus their efforts on improving algorithms controlling non playable characters behavior. The goal is to create NPC which will act as real person. There are several ways to achieve that. One of the ways could be taking closer look at researches touching human behavior. There are plenty of psychological and behavioral theories describing brain reactions reasoning, causing specific behavior. Unfortunately most of those theories cannot be easily implemented because those contain too many abstract assumptions and concepts. On the other hand it can be noticed that most of theories are agreeable about important role of emotions. After consideration of human behavior and decision making, impact of emotions has to be underlined.

Taking a deeper look into most popular agents control techniques, used in games development, led us to simple conclusion that there is no influence of emotions in any of them. This paper researches possibility of enhancing commonly used algorithms by artificial emotions.

There are plenty of psychological models which cannot be adapted for IT usage. By the course of time cooperation of psychologists and computer scientists resulted in researches leading to creation of new theories. Those new models were focused on IT usage. As an example it can be used SIMPLEX (Simulation of Personal Emotion Experience) engine, which simulates experiencing personal emotions as a result of application events. Agents have modelled goals, emotions, mood-states, personality, memory and relationships [5]. SIMPLEX engine is based on theoretical model OCC (name comes from authors surnames: Ortony, Clore and Collins). This emotion theory is based on appraisal and resulted in foundation for artificial emotion systems [5], [6]. Another example can be FAtiMA (Fearnot AffecTIve Mind Architecture) engine which is focused on agent's behaviour, influenced by emotions and personality. FAtiMA also derives from the OCC emotion theory [6]. Its architecture contains planning elements and is based on emotions impact and personality.

5 PSI THEORY OVERVIEW

Dietrich Dörner's Psi theory is characterized by another approach to modeling and emotions impact. Main assumption of this model is to maintain homeostatic balance in dynamically changing environment. Controlled by Psi theory agent contains collection of drives whose main goal is to fulfill them [1]–[4]. When human feel strong need it can be reflected in his behavior. The author made conditional on human behavior creation process from components like: perception, motivations, cognitive process, memory, abilities to learn and emotions. Psi theory because of capability to self-regulation, seems to be a good candidate to enhance artificial intelligence in computer games. That gives the possibility to create non-playable character whose behavior is not scripted linearly. That is the reason why in this paper we will look down deeper into details of main assumptions of Psi theory [1]–[3].

Explicit symbolic representation – in this theory declarative, procedural and tacit knowledge should be represented by hierarchical networks of nodes, where single node contains explicit data, allowing to describe and compare objects and events.

Memory – the Psi theory contains a world model where agent dwells. Current point in time is extrapolated into a situation image which lead to creation of episodic memory. This type of memory can be used to lead out some expectations and predictions. Image composition contains information about encountered objects, executed actions and random events. This component can be used by model in situation analysis and decision making. This component can also be used to keep information about emotional impact caused by specific actions, objects and events.

Perception – in Psi is based on concept of recognizing objects and events by using hypothesis named as HyPercept (hypothesis directed perception). It can be understood as a bottom-up cueing of hypotheses that is interleaved with bottom-down verification. It is universal principle that can be used to recognize objects, events, etc. from all kind of agent sensors.

Urges/Drives – agent's activity is based on collection of predefined urges. Those needs are divided into three groups,depending on their priority for agent. Most important of them is physiological urges group. As the example could be hunger, thirst, safety, etc. Second type are social needs. This group describes urges leading to interactions between agents and other social behaviors, as an example we could use, being willing to help someone in need. Last type describe cognitive urges, which can be explained as a need of expanding knowledge, willingness to getting more information about objects, events and agent's environment. System of urges is most important component in Psi theory, because all activities of this theory comes from needs which value is over the activity threshold, when this happens drive billows into motivation.

Pleasure and Distress – can be described as a signal reflecting change of the agent's demand after taking any action. The strength of pleasure/distress signal is proportional to the change of urges value affected by the action. Signal value can be used as a reinforcement values in learning process.

Modulation – it can be explained as a set of global parameters, touching agent's cognitive processing, which adjust its resources to internal and environmental situation. Modulation component is in charge of controlling behavioral tendencies, choosing and activating specific behaviors, the rate of orientation behavior and the size of activation spreading in perception, memory, planning and decision making.

Emotion – in Psi theory emotions cannot be understood as an independent subsystem, component or set of parameters but a part of whole cognition system. The modulation of perception, behavior and cognitive processing is influenced by emotion. In Dörner's theory can be understood as a set of configurational parameters of the cognitive modulator. Because of in this model we have to focus on primary emotions like: fear, joy, relief, anger or surprise, but not on attitudes like jealousy or envy.

Motivation – motives are combinations between urges and a goal. It can be represented as a situation allowing agent to satisfy his needs. Theory allows to be activated more than one motive, but only one can be chosen to determine agent's behavior. Selection of the dominating motive depends on strength of urge and a probability of achieving the goal. The choice of dominant motive can be regulated by selection threshold parameter which can be case of individual variance and drive strength.

Learning – in Psi theory learning process takes account of hypothesis based perception component and because of that it comprises assimilation or accommodation of perception schemas. Learning process depends on reinforcing the associations of actions and preconditions with goals.

Problem solving – in this model problem solving focus on finding path between current situation and a moment when goal is achieved. Conceptually it is divided into steps: first, problem solving searches for immediate solution, action which fills agent's needs. If there is no result, component searches behavioral routines. Next step rely on searching a chain of actions, where completing a whole plan achieves a goal and reduces corresponding needs. Last option is just in case when all steps fail and it set action for exploring with an eye to finding new solutions. This component is strongly context dependent. Conceptually it is also subject of modulation. According to the Psi theory, most of problem solving algorithms cannot be modeled without some assumptions.

Language – in this theory should be understood as syntactically organized symbols used to designate conceptual representations and a model of language [1]–[3].

6 PSI THEORY ADAPTATION

Mentioned in the section 5 assumptions, forms the Psi theory. Some of them seems to be a little too abstract and need to be specified or modificated in the definite application. Others looks to be simple and reusable.

There are many kinds of players, some of them prefer nice looking games, when for others graphics does not matter, they need a challenge coming from the game. There are many types of players and reasons why they play and what kind of games. However, there is something that all players expect from game – smoothness of working. Computer games are real time programs. They consist of numerous components like: rendering, audio, physics, artificial intelligence, network, etc. Because of games complexity and being real time applications, every component and script have to be efficient. This is the main reason why we will adapt the Psi theory for our purpose. Game artificial intelligence does not have to be that complex and independent as theory assumes. We will focus on modifying only two abstract and composed components, trying to keep from theory as many as possible.

The discussion about adaptation of Psi theory for games purpose, should begin with drives/urges. The idea of needs as a central component, was one of the reasons why we focused on this theory. That concept seems to be natural, when human feels urge, he decides to act. It is called goal oriented behavior and it is known practice. Because of that it we decided to use drives/urges as follows: Agent contains set of predefined needs, where every single urge belongs to one category (physiological, social, cognitive). Every group responds to a drive importance, most important are physiological needs, least group are cognitive urges. This component main task is to measure and keep needs values. If given urge value is over the activation threshold value, model generates a motivation, with strength equal to difference between values. Motivations will get minor change. Condition relating to the creation of the motivation, where urge value exceeds the threshold and there is an opportunity to execute given action, have been simplified. We assume that in game environment there is always a possibility to execute every action. There is still rule allowing to be more than one active motivation, the choice of dominant one is dictated by the strength multiplied by importance. However, because of modifications in other components, choice of dominant motivation can be under the influence of strong emotions and because of that, as a dominant can be picked not the strongest one. At this point we decided to create bidirectional relation between motivations and emotions. It was assumed that when there are couple of strong motivations generated by urges, it can generate negative emotion, because agent is in unpleasant state. As we mentioned emotions. we will describe our idea how to adapt this component. We decided to make emotion two-stage impact. First stage, as it was mentioned above, affects selection of dominating motivation. Second stage, where emotions take part is action selection. When agent have chosen dominating motivation, his goal is to reduce drive, which generates that motivation. At this point, emotions effect is similar to the effect in first step of applying them. High value of negative emotions affects action selection by blurring values important in decision making process. There are other possibilities to use emotions, as a way of making artificial intelligence more human like. Actions aside from influencing drives, can make an impact on emotions too. It can be calculated as additional value fulfilling fake need which is emotion stability. Another idea of using emotions from Psi theory but not affecting this algorithm intrinsically. Emotions can be used to change – behavior controller parameters like movement speed, destination, etc. In theory adaptation we allowed ourselves to create structure like drive and motivation, to hold the dominating motives, it is goal. It is easier to understand that having motives lead agent to create or pick goal. Agent can have multiple motivations but only one goal. In case of computer games, we decided to simplify perception. Now agent will not build network of hypothesis to recognize object or event. Agent will have sensors like sight and hearing and recognition will be simplified to compare data from sensor to agent's knowledge or memory. If object is new for agent it will convey necessary data to agent. As a knowledge or memory we understand set of information required to compare and recognize objects, actions or events. Because of changes in knowledge and perception, language and explicit symbolic representations also got simplified to set of predefined structures. Learning also were adjusted to use in abstract environment and data. In our adaptation of Psi we decided to change the way of learning to, modifying an influence parameters touching objects, actions, etc. These parameters will enhance agent's decision making process. Executed actions will send feedback information about their effect as joy/distress signal.

New component added to theory as an extension is personality. Agent has set of personality traits which modifies actions, objects and events influence on emotions and needs. For example personality can be used to create kind of brave agent who will not be easily affected by negative emotions and his urges will not change so quickly.

Summing up, the Psi theory adapted for computer games agents is composed of set of sensors like sight, hearing, etc., allowing agent to receive information from environment. Data obtained from these sensors will be put under perception structure, which will try to identify object or event base on knowledge and memory. Received impact from recognized object, will be modified by agent's personality traits, which will lead to reinforcement or weakening the influence on agent's drives and emotions. Next stage of model process analyze urges values and compare with threshold. If needed drive generates corresponding motivation. At this point model have to compare all existing motives and with having an eye on emotion state it try to find dominating one to generate agent's goal. When algorithm find most important motive it searches knowledge and memory structures to get information about most efficient action. At this point emotions also put impact on decision making, they can lead to pick wrong action. During execution and after finishing the action, model gets feedback with influence on npc's emotions and drives. This data is also used for learning to enhance the agent's memory and decision making process. It can be easier to understand after taking a look into the Figure 1.

7 TESTING METHOD AND RESULTS

The idea of Psi theory adaptation had to be tested, so it has been decided to create use case where the assumptions could be tested. Agent is placed in simple closed environment where he follows a defined path. Over agent model there is added tag with color corresponding to agent's emotional state. While moving, agent gets under influence of one of two objects, one with positive impact and the other characterized negatively. In the Figure 2 it can be seen testing environment.

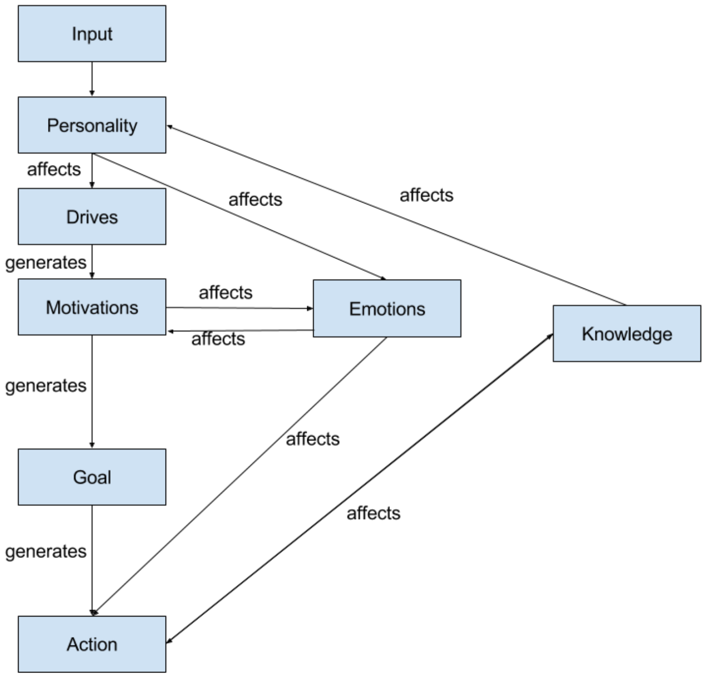


Fig. 1. Psi adaptation data flow diagram

Agent in this demo contains one need an this is safety and one pair of emotions: joy and distress. When joy is dominant emotion, tag over agent comes turns into green color, when opposite and dominant is distress emotion, tag is getting red. Agent's knowledge consist of four elements at this stage:

- Characterized positively object – which increase value of safety to trigger agent action, where expected behavior is to stay as close as possible to decrease urge value and increase joy emotion and reduce distress.
- Characterized negatively object – which increase value of safety to trigger agent action, where expected behavior is to get away as far as possible to decrease urge value and increase distress emotion and reduce joy.
- Accelerate movement – action expected to be executed when safety urge is higher than threshold and agent senses high distress emotion caused by negative object.
- Decelerate movement – action expected to be executed when safety urge is higher than threshold and agent senses high joy emotion caused by positive object.

Scenario of this test is simple and predictable, while moving, when agent face positive object it should slow down to reduce safety and increase positive emotion. With slower speed it still moves away from object. Getting away from object makes agent to return to neutral state, where movement speed is slightly increased, emotions are reduced to zero values and need of safety is slightly increased. Facing negative object makes agent safety increased as a distress value. Expected agent action is to move faster to get away from object faster. When agent drifts away his emotions and needs return to base values.

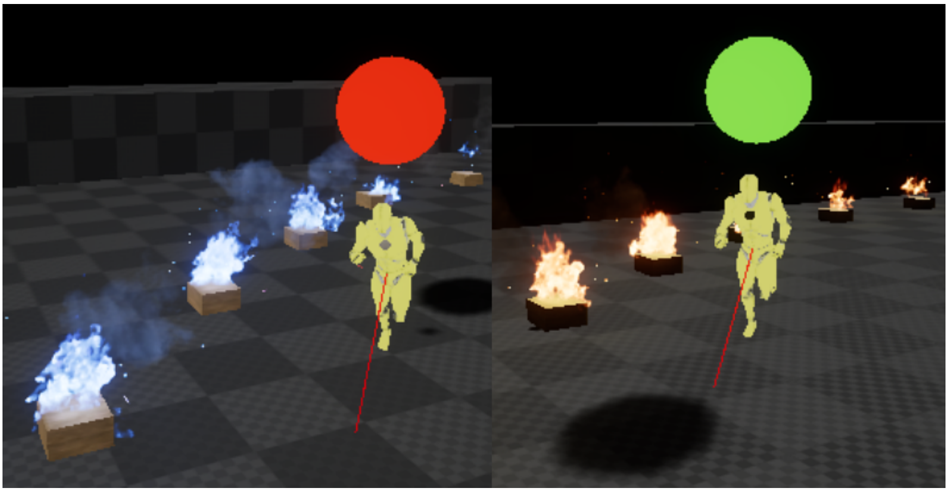


Fig. 2. Screen from testing environment. The left tag is red and the right tag is green

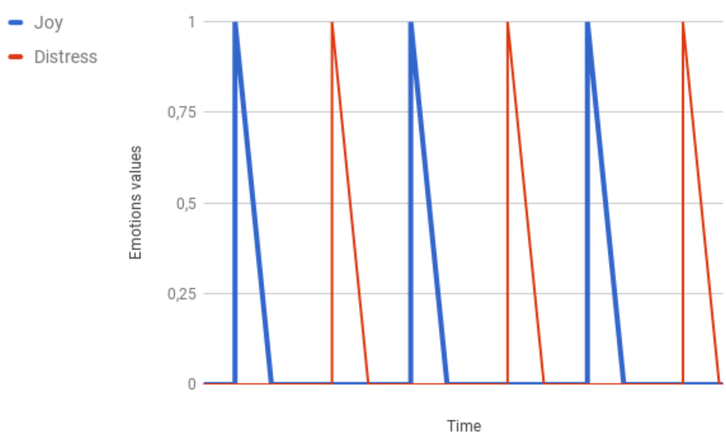


Fig. 3. Joy and distress values received in time period of tests

In the Table 1 there is a sample of data received in testing environment, some indirect values has been ignored to make this table more readable. First column represents Joy emotion and opposite of it – Distress is next to it. Third column represents Safety drive strength. Last value in rows shows movement speed which is conceptualy influenced by emotions. Figure 3 shows joy and distress values in

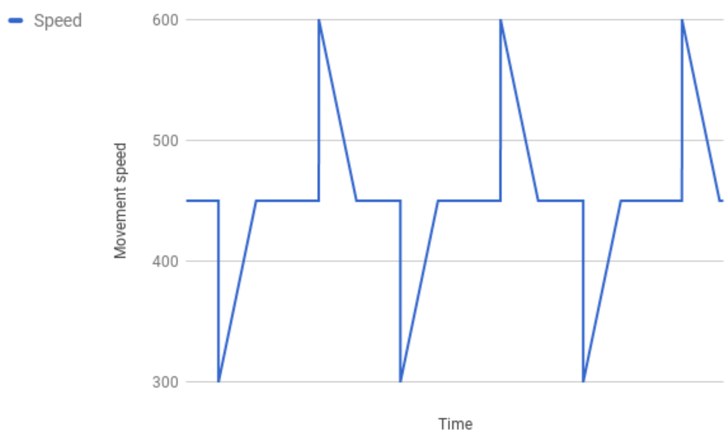


Fig. 4. Character movement speed values received in time period of tests

a time period during tests, where agent met 3 times positive and negative factor. In the Figure 4 it can be seen character movement speed value in corresponding time period. It can be easely notcied that received from tests results covers described above expectations. When joy value in the Figure 3 is rising in the Figure 4 agent's movement speed is falling. On the other hand when agent's distress emotion is getting higher in the Figure 3, it can be noticed reduce of movement speed in the Figure 4.

8 Conclusions

Simple test conditions allowed to check primitive and intuitive drives and emotions dynamics for this adaptation. Therefore, first look on predicted behavior. Results of test present correct influence of basic emotions (positive and negative) and therefore certain stimulus on agents actions. This dependency shows great potential in usage of PSI theory adaptation in decision-making engines for agents. There is still wide space for improvements and enhancing presented mechanism. For example, there are many more different emotions that take part in deliberative process, what should be also considered. However, result of this test seems to be promising for future work.

9 Future work

In next stage of research about Psi theory adaptation and impact of emotions on artificial intelligence in computer games, we are planning to test our model in more complicated scenario. Environment will be expanded on more objects influencing new emotions and drives. Agent knowledge will be more advanced because of new possible actions. In longer future, there is idea of testing Psi adaptation in connection with other artificial intelligence algorithms like behavior trees to make it little more predictable and efficient.

TABLE 1
Some results received in testing environment

Joy	Distress	Safety	Speed
0.0	0.0	0.0	450.0
0.0	0.0	0.0	450.0
0.998	0.0	0.998	300.299988
0.996	0.0	0.996	300.599976
0.944001	0.0	0.944001	308.399902
0.942001	0.0	0.942001	308.69989
0.694004	0.0	0.694004	345.899414
0.692004	0.0	0.692004	346.199402
0.400006	0.0	0.400006	389.999054
0.398006	0.0	0.398006	390.299072
0.282006	0.0	0.282006	407.699097
0.280006	0.0	0.280006	407.999084
0.100006	0.0	0.100006	434.999146
0.098006	0.0	0.098006	435.299133
0.000006	0.0	0.000006	449.999146
0.0	0.0	0.0	450.0
0.0	0.0	0.0	450.0
0.0	0.998	0.998	599.700012
0.0	0.996	0.996	599.400024
0.0	0.826002	0.826002	573.90033
0.0	0.824002	0.824002	573.600342
0.0	0.688004	0.688004	553.200623
0.0	0.686004	0.686004	552.900635
0.0	0.590005	0.590005	538.500793
0.0	0.588005	0.588005	538.200806
0.0	0.446006	0.446006	516.90094
0.0	0.444006	0.444006	516.600952
0.0	0.268006	0.268006	490.200897
0.0	0.266006	0.266006	489.900879
0.0	0.124006	0.124006	468.600861
0.0	0.122006	0.122006	468.300842
0.0	0.000006	0.000006	450.000854
0.0	0.0	0.0	450.0

ACKNOWLEDGMENT

This work was supported by The National Centre for Research and Development within the project "From Robots to Humans: Innovative affective AI system for FPS and TPS games with dynamically regulated psychological aspects of human behaviour" (POIR.01.02.00-00-0133/16).

REFERENCES

[1] Bach, J.; Bauer, C.; Vuine, R.: *MicroPsi: Contributions to a Broad Architecture of Cognition*. In Proc. of KI2006, Bremen, Germany, 2006.
[2] Bach, J.: *Principles of synthetic intelligence. Building Blocks for an Architecture of Motivated Cognition*. PhD Thesis, University of Osnabrück, March 2007.
[3] Bach, J.; Bauer, C.; Vuine, R.: *Modeling Emotion as an Interaction between Motivation and Modulated Cognition*. Workshop on Standards in Emotion Modeling, Leiden, Netherlands, 2011.
[4] Yii Lim, M.; Dias, J.; Aylett, R.; Paiva, A.: *Intelligent NPCs for Educational Role Play Game*. In: Dignum F.; Bradshaw J.; Silverman B.; van Doesburg W. (eds) Agents for Games and Simulations. Lecture Notes in Computer Science, vol 5920. Springer, Berlin, Heidelberg, 2009.

[5] Kessler, H.; Festini, A.; Traue, H. C.; Filipic, S.; Weber, M.; Hoffmann, H.: *SIMPLEX – Simulation of Personal Emotion Experience – A new Model of Artificial Emotions*. ISBN: 978-3-902613-23-3, University of Ulm, Medical Psychology, Institute for Media Computing, Germany, 2008.

[6] Dias, J.; Mascarenhas, S.; Paiva, A.: *FAtiMA Modular: Towards an Agent Architecture with a Generic Appraisal Framework*. In: Bosse T.; Broekens J.; Dias J.; van der Zwaan J. (eds) Emotion Modeling. Lecture Notes in Computer Science, vol 8750. Springer, Cham, 2014.

Patryk Kochański Engineer – graduate of first degree Computer Science on FTIMS division from Łódź University of Technology in Games Technology and Computer Simulations specialization. Participant of United Europe founded project: "Informatyka – kierunek zamawiany na Politechnice Łódzkiej" in module: "Grafika Komputerowa w Rozrywce i Grach Komputerowych". Co-creator of mobile game: "Donut Madness". Participant of many game james like "Slavic Game Jam", "Global Game Jam", "Win 10 hacks" and more. Laureate of "Łódzki Game Jam 2" in 2015 with project: "Bad Day". Co-programmer of games during work in Teyon industry: "Bird Mania Party" released on Wii U, "Planar Conquest" released on PS4 and Xbox One. "Crazy Chicken Director's Cut" released on Google Play.

Marcin Daszuta Ms graduate of the second degree studies in Information Technology at the Faculty of Technical Physics, Information Technology and Applied Mathematics at Lodz University of Technology in specialization Computer Simulation and Games Technology and Ph.D. student at the same Faculty. Laureate of ZTGK tornament in 2013 and participant of Imagine Cup competition in 2015. Co-author of a few scientific publications. Participant of scientific research project "From robot to human, innovative technology allowing to fluent and dynamic regulation of influence cognitive and emotional psychology on actions of AI" as developer. Employee in Ericsson Ltd. company as software developer for 2 years. Interested in computer games, computer grathics, virtual reality, computer stimulations, mobile systems, mobile applications, artificial intelligence, level design in computer games.

Dominik Szajerman was born in Łódź, Poland, in 1977. He graduated in computer graphics and obtained the Ph.D. degree from Department of Technical Physics, Information Technology and Applied Mathematics at Technical University of Lodz, Poland, in 2001 and 2010, respectively. During his pre- and postgraduate experience he was involved in a number of research projects in the areas of computer graphics, computer games and human-computer interaction. His academic interests lie in the computer graphics, GPU processing, computer games development and image processing.

Adam Wojciechowski PhD, DSc, Institute of Information Technology, Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology.

Supervisor of an IT teaching specialisation on Computer Games and Simulations. Since 2016 vice-Dean for Development and Business Cooperation at the Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology. Since 2016 vice-chairman of the GameInn Steering Committee in the National Centre for Research and Development. The Lodz University Rector representative in the Lodz Game Dev Cluster and Faculty Dean representative in the ICT Central Poland Cluster.

Research interests encompass: human-computer interaction, cognitive computing, computer graphics, computer vision, computer games, virtual environments, computer simulations and visualizations, computer animations, biometry and machine learning. Author or co-author of over 50 research papers. Coordinator of 3 and participant of 13 national and international research and development projects, financed by Polish and European programs.

Optical flow as a sense of artificial intelligence in computer games

Damian Pęszor*, Agnieszka Szczęsna†

*Research and Development Center, Bytom, Polish-Japanese Academy of Information Technology, Warsaw, Poland, damian.peszor@pja.edu.pl,
†Institute of Infomatics, Silesian University of Technology, Gliwice, Poland
agnieszka.szczesna@polsl.pl

Abstract

The article deals with the issue of limited knowledge at disposal of artificial intelligence agent in the context of computer game or simulation. Replacing an absolute knowledge about the scene with a perception of it using a sensory faculty which provides knowledge corresponding to human perception allows bringing an artificial intelligence agent closer to a player. Such agent is able to make mistakes similar to those resulting from human imperfection but avoids issues related to limiting absolute knowledge because of the necessity of being able to be beaten by a human enemy. The article uses a sense in form of optical flow calculated by a robust eFOLKI algorithm as an input for an artificial neural network which uses complex-valued input. The network is taught by back propagation to solve a task of predicting the correct tilt and pan necessary for a virtual arrow to hit the approaching enemy in an environment based on popular Minecraft game.

Index Terms

optical flow, artificial intelligence, computer games, computer graphics

1 Introduction

As technology supporting complex computer games evolves, characteristic of those games evolves as well. In place of two-dimensional mazes, three-dimensional environments emerged, which are as complex as real locations, which they simulate. Aside from the graphical quality of the presented world, it is essential for the player that the environment will be as functional, as it is realistic in appearance. While this quality, as expressed by a complexity of possible interactions between the player character and virtual environment, can be very refined, the interaction between the agent of artificial intelligence and player character is often limited by perception of said agent.

In the classical approach, virtual enemy placed within such an environment perceives only a fraction of it. Geometrical representation of the world, often simplified, allows navigating through the scene while avoiding obstacles. Ray casting decides whether player's character is visible from the position of the agent. Knowledge of the real position of player's character, events he triggers and state flags allow for reacting on changes within environments which might be a result of player character's actions. Scripts and position-based triggers allow for hiding behind some part of the landscape in order to avoid detection, enemy gunfire and similar dangers.

Development of artificial intelligence present in computer games (as well as simulations and training applications) is directed mostly towards the creation of a demanding opponent. Winning with such an opponent is supposed to be a sign of the player being more competent in the field of the game than intelligence which is artificial in nature, but still similar to his own in terms of specific game mechanics. Not human, but human alike. The distinct difference in a way artificial intelligence agent and human player are perceiving the world is therefore in direct opposition to the goals that artificial intelligence creators are trying to achieve. The agent is equipped with additional information – functionally similar to human senses – which make it impossible for a player to use tactics which he understands and could successfully employ in real life. Let us present an example. One could try to deceive artificial intelligence agent by using colour-based camouflage. A tactic which is well-known and successfully used both in nature as well as in the military. Since agent of artificial intelligence often uses ray casting rather than analysis of colour, such tactics will simply not work. On the other hand, a player might deceive agent in ways that exploit limits of its perception, for instance, it is often that artificial intelligence agents have a range of their vision, which allows a player's character to position himself in clearly visible space, but outside agent's range and thus remain undetected. Such range is however needed for an agent to be unaware of player's character presence until the player is ready to face the danger.

Analysis of image sequence as rendered from the perspective of artificial intelligence agent allows the detection of movement and tracking of objects, which in turn allows perceiving the environment in a way more similar to human. One of the most important families of algorithms of such vision-based system is known as optical flow. It is widely used in terms of, detection of movement by cameras of alarm systems, detection of obstacles on unmanned aerial vehicle's trajectory or land-based autonomous platforms [1]–[3] and many different applications. It can potentially also be used by agents of artificial intelligence to detect, track and aim at a player's character.

The authors propose using analysis of optical flow in order to equip agent of artificial intelligence into a sense which is highly receptive in terms of dynamic changes in scene and partially replacing perception based on data of other types than used by a player.

2 Related work

One of the most promising projects related to usage of artificial intelligence in games, wherein artificial intelligence takes on the role of a player is Atari Grand Challenge [4]. The dataset associated with the challenge contains large amount of replays collected from Atari 2600 games. While certainly related to the issue, one has to understand that it puts artificial intelligence in the same role as the player, not as an agent which can work against the player. While this might be possible within certain games presented within the challenge, its usage as a tool to develop intelligent enemies within the game framework is limited.

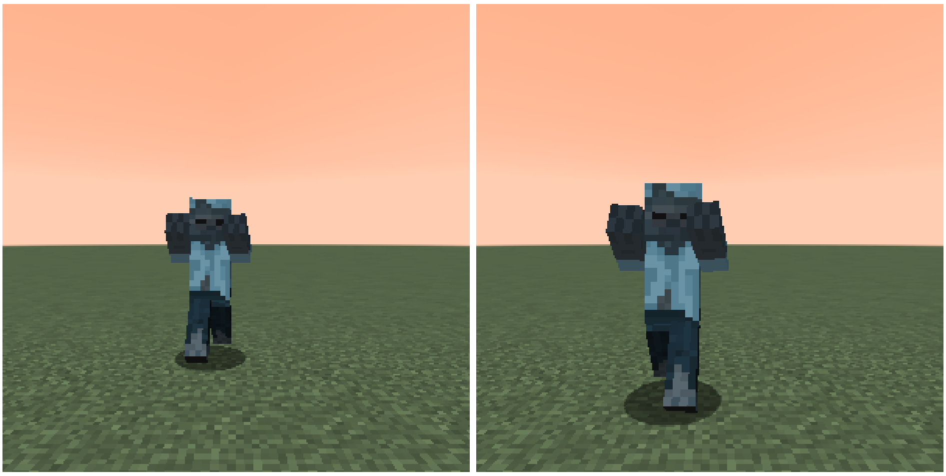


Fig. 1. Two consecutive frames recorded from Microsoft Project Malmö environment. Both 300 pixels in both width and height. A creature approaches the agent from a randomly selected position within the area inside of the field of vision of the agent

The recent introduction of Microsoft Project Malmö (see section 3.1) allowed for research focused on the tools available through this platform. Video obtained through Microsoft Project Malmö was used to solve different tasks. Geiger et al. consider using video output from Microsoft Project Malmö in three missions where agent's goal is to find the specific target [5]. The paper describes thoroughly a datagrid-based approach, one that uses absolute information rather than agent's own perception to teleport around the grid.

Allison et. al introduce the "Wizard of Oz" agent which observes young Minecraft players' actions and tries to help them in their tasks by first understanding them. Authors indicate that agent is using same visual input as a player does [6].

Monfort et al. use the artificial neural network on the basis of video output obtained through Microsoft Project Malmö in order to navigate a maze of random blocks [10]. While the task is different, and they do not use optical flow in their navigation, the usage of an artificial neural network makes it similar to our approach, and generally, they do use visual input as sensory faculty.

A step closer to our approach is the work of Udagawa et al. [11]. They use visual input obtained from Microsoft Project Malmö as sensory faculty to attack creatures, as in our example described in section 3.1. The key difference is that they do not use a ranged attack, so the aiming is not that important and difficult and the fact that they do not calculate optical flow, but rather work directly on the visual input.

A task closer to presented in this paper is used in terms of artificial intelligence agents designed for the purpose of first person shooter games. An example of those might be found in [12], which presents Clyde, the agent used to play popular DOOM game. While achieved differently, the goal to aim correctly is similar.

Justesen et al. review advances in artificial intelligence playing computer games and using artificial neural networks [7]. Various genres of computer games and a thorough comparison of network architectures make this position worthwhile. Microsoft Project Malmö is also considered.

Another recent review of artificial neural networks' architectures is presented in [8] by Oh et al. This work is focused on their applicability to Minecraft and therefore is directly related to the issue at hand.

Tessler et al. consider yet another challenge within the framework of Minecraft and Reinforcement Learning [9]. Namely, the paper tackles the issue of sharing knowledge gained by artificial intelligence agent during between different tasks that it performs, which is important in terms of missions that are composed of many tasks such as the one described as an example in this paper.

It is also worth to mention a recent work [13] on developing an integrated framework for artificial intelligence experimentation which seems to fit the context of providing artificial intelligence agents with sensory faculty very well.

3 Materials and methods

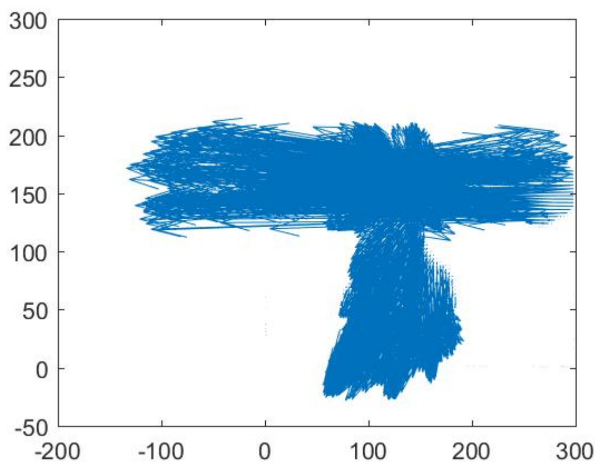


Fig. 2. Quiver plot of optical flow vectors obtained from consecutive frames as presented in Fig. 1. Note that vectors are presented in the space of the preceding frame, as such some vectors point outside of 300 by 300 resolution of the frames

3.1 Microsoft Project Malmö

Microsoft Project Malmö [21] is an experimentation environment built around popular computer game Minecraft and provided by Microsoft. It is designed to allow artificial intelligence-related research using a rich, interactive environment provided originally for Minecraft player. An artificial intelligence agent is able to perform all tasks that player could originally do; move through three-dimensional space, attack various creatures, use objects, gather resources, build constructions and even create logic circuits among other things.

Absolute knowledge can be obtained from Microsoft Project Malmö, for example in form of positions of agent and other objects and creatures on the discrete grid. Truly useful data, however, can be obtained in form of video frames as seen by artificial intelligence agent in real time. This allows calculation of optical flow between consecutive frames and thus utilising such information as a sensory faculty for artificial intelligence agent is possible. A pair of sample consecutive frames is presented in Fig. 1. Note that in this case, for the sake of simplicity, an empty, flat world with no weather conditions has been generated.

In the presented case, the agent has to deal with the approach of a hostile creature. As mentioned before, artificial intelligence agent in Microsoft Project Malmö is able to use all the actions available for a player. This includes using bow and arrows, which we equipped the agent with. When the agent will notice movement, he has to activate bow, wait for a second to full activation and release the arrow in the direction he is facing. During the activation time, the agent can adjust his orientation through pan and tilt. The goal is, therefore, to use optical flow to determine the pan and tilt that will result in arrow hitting the target. We spawn the creature at various places far enough for the agent to attack it at least once before the creature will be close enough to attack the agent. Note, that the target is moving and the arrow flight is not instantaneous. This means, that the artificial intelligence has to take further movement into account and predict the correct orientation. This is exactly the task that the player has to undertake in this kind of situation in original Minecraft game.

3.2 Optical flow

Optical flow describes relative motion of the agent and objects in the world the agent perceives. It is expressed through two-dimensional vectors describing the translation in screen space between projections of visible points in the scene through the time which passed between consecutive frames. It is worth to note that optical flow in itself does not describe the change in three-dimensional scene (although one can estimate such change on the basis of optical flow using shape from motion algorithms), but rather the perception of such changes. This perception comes from perspective projection which is natural for observer, whether it is human or a human-like artificial intelligence agent. This is actually an advantageous feature in terms of artificial intelligence since it naturally models the influence of magnitude and direction of movement into the degree of noticeability of the movement for given observer.

A key stage in the calculation of optical flow is finding correspondence between pixels in consecutive frames. Although application which contains three-dimensional environment can potentially provide all data needed to establish such correspondence, we advise against using such an information. The colour-based similarity between pixels in neighbourhood allows for incorrect identification of correspondences, which models human insensitivity on small changes in the similarly coloured area. This allows for colour-based camouflage to be a vital part of gameplay as well as directing the attention of the agent by introducing moving (e.g. thrown) objects.

Calculating optical flow for an image is a relatively expensive operation, which restricts the class of optical flow algorithms that can be used in the context of the real-time responsive application, to those, which sacrifice the accuracy of movement prediction to obtain a short time of calculations. KITTI database [14], [15] offers a comparison of available algorithms for short time of processing. At the moment of writing this article, four algorithms meet the requirements for time of processing to be shorter than 0.1s. Namely, the anonymous and unreferenced PWC-Net and Unsup Flow algorithms, Dense Spatial Transform Flow (DSTFlow, [16]) and Prediction Correction Optical Flow + Adaptive Coarse-to-fine-stereo algorithm (PCOF+ACTF, [17]). It is worth to note that authors decide to use the PCOF+ACTF algorithm for the purpose of research around usage in the context of real-time artificial intelligence. While this algorithm is fast enough for real-time, it obtains such a characteristic at the cost of accuracy. This results in vectors as presented on quiver plot in Fig. 2. The magnitude of those vectors is presented in Fig. 3 and the direction in Fig. 4. Note that this information still has to be processed in some way in order for the agent to react correctly to the perceived change in environment.

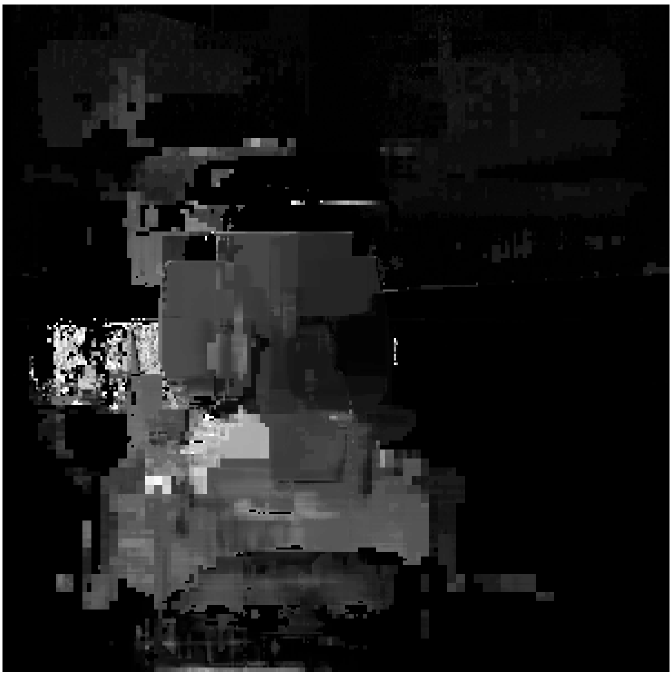


Fig. 3. The magnitude of optical flow vectors obtained from consecutive frames as presented in Fig. 1. Black represents the magnitude of 0, while white represents a magnitude of 128 pixels or above. Note, that the magnitude is presented in the frame of the preceding image (left one in Fig. 1)

The point of using optical flow is to use artificial intelligence agent's perception instead of absolute knowledge about the environment the agent is in. Such information, however, might be used in order to restrict the analysed part of visible space and to eliminate noise coming from agent's own movement. This allows increasing the efficiency of algorithm calculating optical flow while at the same time still restrict the absolute knowledge about the scene on part of the agent. It is worth to note, that due to the simplicity of presented scene in the example provided within this article, such performance-related approach was not needed. However, developers of application aimed toward computer game market would find such an option tempting.

$$\begin{pmatrix} u_{pred} \\ v_{pred} \end{pmatrix}(x,y) = \Pi(RX_t(x,y,d_t)+T) - \begin{pmatrix} x \\ y \end{pmatrix} \tag{1}$$

The prediction stage, presented in Eq. 1 as described in [17], is responsible for calculation of the part of optical flow (u_{pred}, v_{pred}) for given (x, y) coordinates which results from the motion of agent itself in static environment. In PCOF+ACTF, the motion components, that is rotation R and translation T of an agent, are obtained using efficient Visual Odometer eVO as proposed in [18]. In the context of artificial intelligence in a computer game, one does not need to calculate them in such a matter, as they can be obtained directly from game logic or physics system. Using data derived from visual input might introduce additional noise from the imperfection of motion estimation, which is one way of modelling human imperfection in recognizing player's own movement while playing a video game. This is, however, an inconvenient way of modelling such a phenomena, especially in presence of other sources of noise, when there is a movement on observed scene. The $X_t(x, y, d)$ expression is the triangulation function at frame t used in PCOF+ACTF as presented in Eq. 2. (X_0, y_0) are the coordinates of the optical centre's projection and f stands for camera focal in pixels. d_t is for disparity map's value at frame t obtained using stereo cameras. In case of virtual reality-based games, this could be beneficial as two virtual cameras model the depth perceived by the user of virtual reality headset. In the context of a single screen, however, one can again simply use three-dimensional coordinates X_t obtained directly from game systems.

$$X_t(x,y,d) = -\frac{b}{d(x,y)}(x - x_0, y - y_0, f) \tag{2}$$

For non-virtual reality context the ACTF part of the PCOF+ACTF approach can therefore be ignored. PCOF defines its correction stage as presented in Eq. 3. The crucial part of the algorithm is therefore the exact optical flow algorithm used to detect the part of optical flow that is not based on agent's own movement. For such dynamic environments, [17] proposes the FOLKI algorithm presented in [19]. In the context of this study, a more robust and faster method, the extended FOLKI (eFOLKI, [20]) was used.

$$\begin{aligned} u(x,y) &= \delta u(x,y) + u_{pred}(x + \delta u(x,y), y + \delta v(x,y)) \\ v(x,y) &= \delta v(x,y) + v_{pred}(x + \delta u(x,y), y + \delta v(x,y)) \end{aligned} \tag{3}$$

3.3 Decision-making process

For the purpose of this paper, we assumed that optical flow will be directly analysed using an artificial neural network. While other methods might be used, we wanted a general method, which can easily be used for more difficult tasks than one presented here as an example. Neural networks were previously used in such a direct way for optical flow analysis, e.g. in [22]. As in the mentioned article, we used complex back propagation originally developed by Nitta and Furuya in [23] and as presented in [24]. The representation of the vector as complex input is crucial. In case of ordinary, real, multi-valued unit, the relationship between magnitude and direction of the vector does not affect the internal workings of the artificial neural network as

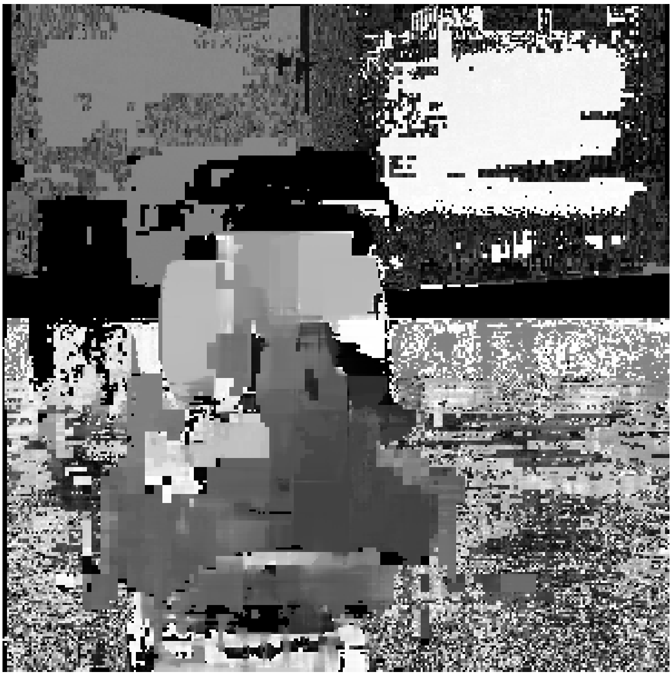


Fig. 4. The direction of optical flow vectors obtained from consecutive frames as presented in Fig. 1. Black represents the angle of 0 radians, while white would represent an angle of 2π radians. Consider Fig. 3 to find areas of interest, as an angle of a vector of low magnitude can be considered just a noise

much and results in different characteristic. An activation function based on complex numbers allows for better modelling of the dependence of direction on the magnitude because, with lower magnitudes, the direction is often just a noise. Input layer consists of one unit per input optical flow vector, which within presented example constitutes a 300 by 300 grid, as such image resolution was used. After consulting [25] and further experiments, we decided to use two hidden layers of 150 and 30 nodes respectively. Output layer consists of just two real units, one responsible for pan and one for tilt. A careful reader might notice that the output could be represented by a single node using the fact, that the complex number consists of both real and imaginary dimensions. In fact, the difference is just a matter of representation rather than actual calculations. The schema of the used network is presented in Fig. 5. After each shot, it is determined whether the creature was hit or not. A successful hit is considered as a positive result, while a miss is considered a negative result. Artificial neural network is therefore taught on the basis of each shot rather than entire mission. Since each shot is performed after the minimum time required for agent to fully draw the bow, the number of frames that can be used as an input for given shot is limited. This means, that each subsequent shot is composed of multiple frames and thus – multiple outputs from artificial neural network. The success or failure information is therefore used for the entirety of the calculations performed for single shot.

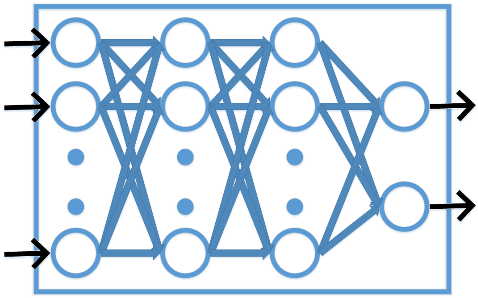


Fig. 5. Visualisation of artificial neural network used in the presented example. The neural network consists of four fully connected layers. The input layer accepts complex numbers representing magnitude and direction of each vector of optical flow. In the presented example, images of resolutions of 300 by 300 are used and therefore as many vectors and input nodes are present. The hidden layer consists of 150 and 30 nodes respectively, while output layer consists of just 2 nodes, one for pan and one for tilt

3.4 Hardware used

Since the calculation of optical flow is relatively demanding and artificial neural network, while fast to use, might be slow to teach when it has a great number of nodes, it is worth to note what kind of equipment was used in the teaching process. Since both optical flow and neural network for such a number of nodes are highly parallelizable, we decide to use Graphics Processing Unit for those tasks. Microsoft Project Malmö uses Minecraft, which is not very graphically demanding, so the Graphics Processing Unit, in this case, can be used for the purpose of calculations without degrading the amount of generated frames per second for the agent to use. The computer used was equipped with the following:

- Intel Core i7-6850K CPU, which has 6 cores working within Hyper-Threading Technology at frequency of 3.6GHz.
- 16GB of RAM.
- NVidia GTX GeForce 1080 Graphics Processing Unit, which is built using Pascal architecture. The GPU is equipped with 2560 CUDA cores, memory of 8GB GDDR5X with bandwith of 30 GBps, working with frequency of 1607 Hz.

4 Results

We recorded the state of the artificial neural network after fulfilling each 50 missions. It is important to note, that a mission will contain a different number of shots fired by artificial intelligence agent due to the fact, that the creature is spawned randomly. The creature orientation is random as well as its position, which is limited to certain area within agent's field of view. This is necessary in order to avoid neural network learning where to shoot on the basis of creature's path rather than optical flow generated by movement of said creature. We provide the number of shots fired within Table 1 after each hundred missions. We tested developed artificial neural network using states obtained after each 50 missions outside of the learning pipeline on 100 validation missions. The accuracy of results can be read from Table 1 and Fig. 6. Note that since initial orientation and position of creatures within validation set was determined randomly (in the same fashion as in the learning set), the obtained 100% accuracy does not mean, that every possible mission is guaranteed to be fulfilled with such an accuracy.

TABLE 1
Accuracy of artificial intelligence agent after given number of performed missions and shots fired

Mission count	Shots fired	Accuracy rate
After 100 missions	342	24.3%
After 200 missions	691	30.1%
After 300 missions	985	44.7%
After 400 missions	1292	58.0%
After 500 missions	1604	61.1%
After 600 missions	1945	72.5%
After 700 missions	2301	88.8%
After 800 missions	2683	93.1%
After 900 missions	3003	96.6%
After 1000 missions	3357	100.0%

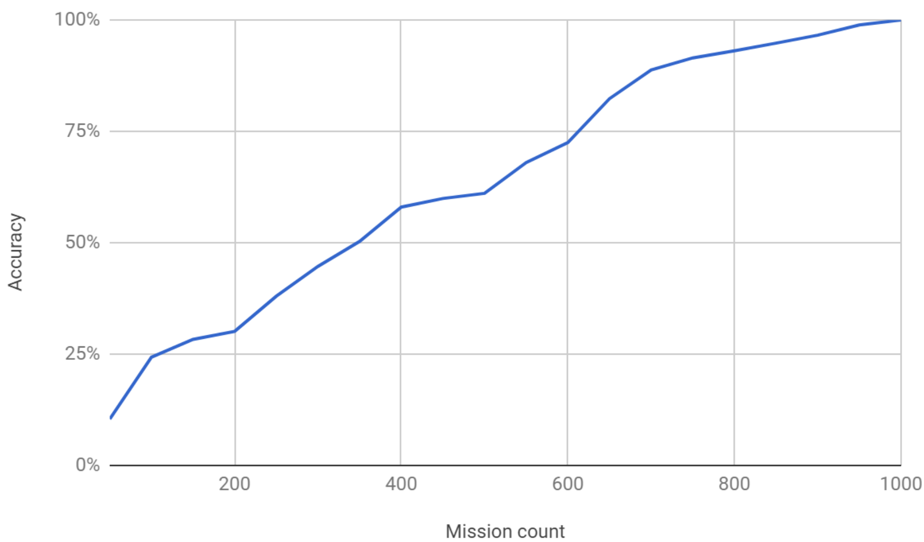


Fig. 6. Chart of accuracy with respect to the number of missions artificial intelligence agent undertook. Data points are located after every 50 missions

5 Discussion

The 100% of accuracy obtained by artificial intelligence agent proves that task presented in this paper is quite easy to solve using the artificial neural network and optical flow. This suggests, that optical flow can be successfully used as a sense of artificial intelligence agent. One might wonder why so many iterations were needed to obtain 100% accuracy within this task. Few elements have to be considered:

- The creature is spawned within the field of vision of artificial agent. Sometimes, the initial rotation of the creature is difficult to handle properly by neural network.
- The creature will sometimes move outside the vision of the agent. In some of the tests, an agent might overshoot his rotation and loose the sight of the creature for good.
- The creature is not entirely solid, it is animated and therefore generates optical flow which can be confusing for neural network.
- Whenever agent rotates in order to aim for the creature, it generates optical flow due to its own rotation. This can be misleading.
- We noticed, that as the neural network is taught, it produces more subtle rotations. This leads to minimization of optical flow generated by agent's rotation and allows for better aiming.
- One might notice, that it is enough to aim into a pixel with the high optical flow. In our approach, however, optical flow is calculated on the previous frame, so the agent has to aim toward the pixels pointed by optical flow rather than origins of those vectors.
- Note that creature is moving and arrow shot by the agent does not travel instantaneously. This means, that neural network actually has to predict where to shoot rather than just shoot into any pixel where optical flow points to.

6 CONCLUSIONS

Presented approach to solving the issue of available information for artificial intelligence brings modern game development closer to closing the gap between agent and player, which in authors' opinion leads to greater immersion, more effective training and more exciting gameplay.

Although using optical flow might substantially affect performance in case of graphically demanding games, the rapid development of technologies which support such games indicates, that the issue is irrelevant. The increase in a number of Graphics Processing Unit's cores allows for high parallelization of computations which is perfect for dense optical flow calculations, especially with algorithms designed for time performance rather than accuracy, as in case of extended FOLKI. It is also worth to note, that computer games that are not developed in order to appear realistically offer such possibilities even at the moment of writing this article.

ACKNOWLEDGEMENTS

This work was supported by Young Scientist project BKM-509/RAU2/2017, task 7 "Metody i zastosowania algorytmów przetwarzania ruchu" ("Methods and applications of motion processing algorithms"), subtask "Stereowizja i przepływ optyczny w wykrywaniu ruchu na scenie" ("Stereovision and optical flow in motion detection in the scene") of Silesian University of Technology, Institute of Informatics, Gliwice, Poland.

REFERENCES

[1] Bonin-Font, F.; Ortiz, A.; Oliver, G.: *Visual navigation for mobile robots: A survey*. Journal of intelligent and robotic systems 53.3, pp. 263, 2008.

[2] DeSouza, G.N.; Kak, A.C.: *Vision for mobile robot navigation: A survey*. IEEE transactions on pattern analysis and machine intelligence 24.2, 2002, pp. 237–267.

[3] Aguirre, E.; González, A.: *Fuzzy behaviors for mobile robot navigation: design, coordination and fusion*. International Journal of Approximate Reasoning 25.3, 2000, pp. 255–289.

[4] Kurin, V.; Nowozin, S.; Hofmann, K.; Beyer, L.; Leibe, B.: *The Atari Grand Challenge Dataset*. arXiv preprint arXiv:1705.10998, 2017.

[5] Geiger, P.; Hofmann, K.; Schölkopf, B.: *Experimental and causal view on information integration in autonomous agents*. Proceedings of the 6th International Workshop on Combinations of Intelligent Methods and Applications (CIMA 2016), pp. 21–28, Ed. Hatzilygeroudis, I., Palade, V.

[6] Allison, F.; Luger, E.; Hofmann, K.: *Spontaneous Interactions with a Virtually Embodied Intelligent Assistant in Minecraft*. Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, CHI EA '17, ACM, pp. 2337–2344.

[7] Justesen, N.; Bontrager, P.; Togelius, J.; Risi, S.: *Deep Learning for Video Game Playing*. arXiv preprint arXiv:1708.07902, 2017.

[8] Oh, J.; Chockalingam, V.; Singh, S.; Lee, H.: *Control of memory, active perception, and action in minecraft*. arXiv preprint arXiv:1605.09128, 2016.

[9] Tessler, C.; Givony, S.; Zahavy, T.; Mankowitz, D.J.; Mannor, S.: *A deep hierarchical approach to lifelong learning in minecraft*. arXiv preprint arXiv:1604.07255, 2016.

[10] Monfort, M.; Johnson, M.; Oliva, A.; Hofmann, K.: *Asynchronous Data Aggregation for Training End to End Visual Control Networks*. Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, International Foundation for Autonomous Agents and Multiagent Systems, May 10, 2017, pp. 530–537.

[11] Udagawa, H.; Narasimhan, T.; Lee, S.-Y.: *Fighting Zombies in Minecraft With Deep Reinforcement Learning*. Technical report, CS229 Machine Learning, Autumn 2016, Stanford University.

[12] Ratcliffe, D.; Devlin, S.; Kruschwitz, U.; Citi, L.: *Clyde: A deep reinforcement learning DOOM playing agent*. Workshops at the Thirty-First AAAI Conference on Artificial Intelligence What's Next For AI In Games, 2017.

[13] Moulin-Frier, C.; Puigbò, J.-Y.; Arsiwalla, X.D.; Sanchez-Fibla, M.; Verschure, P.F.M.J.: *Embodied Artificial Intelligence through Distributed Adaptive Control: An Integrated Framework*. arXiv preprint arXiv:1704.01407, 2017.

[14] Menze, M.; Geiger, A.: *Object Scene Flow for Autonomous Vehicles*. Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[15] Menze, M.; Heipke, C.; Geiger, A.: *Joint 3D Estimation of Vehicles and Scene Flow*. ISPRS Workshop on Image Sequence Analysis (ISA), 2015.

[16] Ren, Z.; Yan, J.; Ni, B.; Liu, B.; Yang, X.; Zha, H.: *Unsupervised Deep Learning for Optical Flow Estimation*. Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17).

[17] Derome, M.; Plyer, A.; Sanfourche, M.; Le Besnerais, G.: *A Prediction-Correction Approach for Real-Time Optical Flow Computation Using Stereo*. German Conference on Pattern Recognition, 2016, pp. 365–376.

[18] Sanfourche, M.; Vittori, V.; Le Besnerais, G.: *eVO: A realtime embedded stereo odometry for MAV applications*. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2013, pp. 2107–2114.

[19] Le Besnerais, G., Champagnat, F.: *Dense optical flow by iterative local window registration*. IEEE International Conference on Image Processing, 2005. ICIP 2005. Vol. 1, pp. 137–140.

[20] Plyer, A.; Le Besnerais, G.; Champagnat, F.: *Massively parallel Lucas Kanade optical flow for real-time video processing applications*. Journal of Real-Time Image Processing, April 2016, Vol. 11, Issue 4, pp. 713–730.

[21] Johnson, M.; Hofmann, K.; Hutton, T.; Bignell, D.: *The Malmo Platform for Artificial Intelligence Experimentation*. Proc. 25th International Joint Conference on Artificial Intelligence, Ed. Kambhampati S., pp. 4246. AAAI Press, Palo Alto, California USA. https://github.com/Microsoft/malmo.

[22] Miyauchi, M.; Seki, M.; Watanabe, A.; Miyauchi, A.: *Interpretation of Optical Flow through Neural Network Learning*. Proceedings of IAPR Workshop on Machine Vision Applications, Tokyo, Dec., pp. 523–528.

[23] Nitta, T.; Furuya, T.: *A Complex Back Propagation Learning*. Trans. Information Processing Society of Japan, Vol. 32, No. 10, pp. 1319–1329 (1991).

[24] Hirose, A.: *Complex-valued Neural Networks: Theories and Applications*. World Scientific 2003, ISBN: 978-981-4485-37-1.
[25] Stathakis, D.: *How many hidden layers and nodes?*. International Journal of Remote Sensing, Vol. 30, No. 8, pp. 2133–2147, 2009.

Damian Pęszor obtained his BSc in Computer Sciences in 2011 at Polish-Japanese Academy of Information Technology and a MSc in same discipline in 2012 at Silesian University of Technology in Gliwice, Poland. He is currently a PhD student at Silesian University of Technology in Gliwice, Poland. His research is currently centered on the development of performance capture-based facial animation, motion analysis and computer vision.

Agnieszka Szczęsna is an adjunct professor at Institute of Informatics in Silesian University of Technology, Gliwice, Poland. She also works closely with Research and Development Center of Polish-Japanese Academy of Information Technology in Bytom, Poland. Her research interests are focused in the fields of motion analysis, geometric modelling and serious games. Since 2012 she is in charge of “Interactive 3D Graphics” specialization of master’s degree studies which is focused on video games design and development.

Impact of backward rotational velocity on menu navigation in virtual reality

Jarosław Andrzejczak, Jan Bełczącki

Lodz University of Technology, Łódź, Poland
jaroslaw.andrzejczak@p.lodz.pl, jan.belczacki@o2.pl

Abstract

Despite fact that virtual environment systems have a very long history, there are still place for improvements and innovations. One of them concerns graphical user interface design. After all these years, many ideas were implemented with different results. Many efforts have been made to let user enjoy the specific system even more. Other branch of VR research and development is based on navigation and reorientation techniques implemented mostly for movement in Virtual Reality space. This paper seeks to investigate the impact of one branch on another by giving a simple task on a ring menu together with proposing an additional backward rotational velocity parameter. Usability tests described in this paper showed that using this parameter increases the task performance time, task efficiency and user satisfaction score for menu navigation in VR. Additionally, article presents comparison of backward rotational velocity parameter to rotational gain parameter as well as tests with both of this parameters and proposition for default values for them.

Index Terms

virtual reality, user interface, navigation

1 Introduction

Despite fact that virtual environment systems have a very long history, there are still place for improvements and innovations. One of them concerns graphical user interface design. After all these years, many ideas were implemented with different results. Other branch of VR research and development is based on navigation and reorientation techniques implemented mostly for movement in Virtual Reality space. This techniques are widely used in situations when VR developers want to limit the real space, constraining it to the tracking area, but don't want to show those limitations in virtual environment [1]. One of these techniques is called "rotational gain" [2]. It enables increasing the rotational velocity in virtual environment, so the user can actually rotate his head with smaller angle [5]. While navigating through virtual space is done mostly using external controllers [16] selecting menu item can be successfully made pointing it using eye gaze and head rotation.

In this paper authors propose the model for studying impact of this reorientation technique on navigating through menu in virtual environment. Authors also propose backward rotational velocity parameter[1] along with its usability test with users.

The contributions to graphical user interface design for virtual reality systems research presented in this article are:

- Increase of menu operating speed by adding reorientation parameter for environment itself.
- Proposition for additional parameter (backward rotational velocity) that affects a backward rotational velocity of menu.
- Proposition for default values of both these parameters (rotational gain and backward rotational velocity) to ensure the fastest and most comfortable experience.
- HCI and usability tests to verify usefulness of proposed as well as original test system created for that purpose.

We start with a discussion of related work in the next section. This is followed by a description of a main idea of the proposed backward rotational velocity parameter. We then present the test system and evaluation stages construction for HCI evaluation and usability tests with users. Next, test results and their discussion are presented. Finally, ideas for further development and final conclusions will be given.

2 STATE OF THE ART

"Rotational gain" technique, enabling increasing rotational velocity in virtual environment can be implemented using following equations [10]:

$$q_d = q_c^k, \tag{1}$$

$$q_d = (q_c q_0^{-1})^k q_0, \tag{2}$$

Equation 1 presents scaling device orientation q_c using k parameter. In case when $k > 1$ we have to do with upscaling, when $0 < k < 1$ – downscaling. Equation 2 introduces initial rotation, so this equation presents a single device rotation from certain zero orientation q_0.

Another interesting method of developing reorientation technique is presented in [2]. For creating 360 degree video system, authors decided to implement dynamic rotation gain based on corresponding head/neck rotational velocity, together with constant, presented in this paper. Figure 1 shows the differences between three panning methods.

This solution gives promising results after pilot studies, to quote the authors: "the user will unknowingly use less physical movement than needed, yet maintain reasonable spatial understanding with higher usability". However, this method needs to be further tested with more formal experiments and wider range of parameter values.

Other methods can consist in using inprecision of human perception (Redirected Walking method [7]), modifying expected movement path (Motion compression method [9]) or using certain orientation point that sidetracks user from working reorientation [6]. However all these methods concerns movement in large virtual spaces.

[1] Backward rotational velocity parameter is referred as μ further in the article.

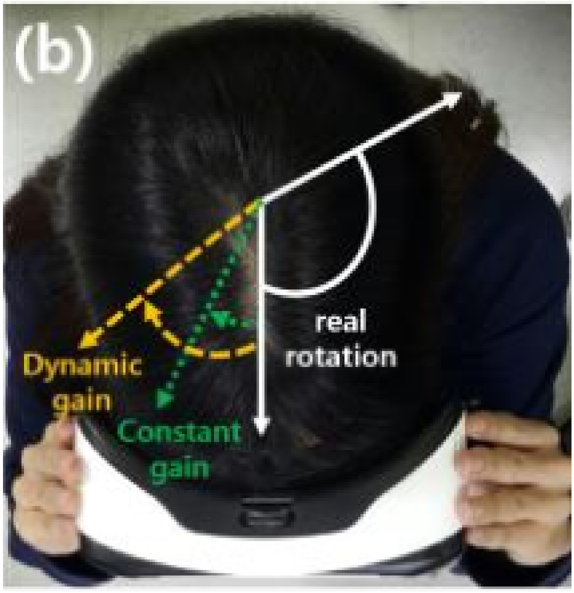


Fig. 1. Three panning methods. Without any gain (solid line), with constant gain (dotted line), with accelerated gain (dashed line). Source: [2]

There are also plenty of methods for navigating through user interface in virtual reality applications by pointing at certain objects. Most of them are presented in [10]. Except for raycasting system used in this paper, authors of this book present some sort of alternative for simple single ray technique called "flashlight".

This method imitates pointing at menu elements using flashlight, which makes selecting much easier but generates a problem which had an impact on final decision. This problem lies within disambugation of desired object in case of multiple objects inside the spotlight. This situation would appear very often for ring menu implemented in our solution, because the elements need to be quite close to each other. The modification called "aperture" presented in [14], introduces controlling the spread of the selection volume (Figure 2). However, this option needs additional buttons for input system and this could make our solution too complicated.

The idea of ring menu has been very well described in [11] and furtherly developed in [8]. Essentially, the elements appear in 3D space around the certain object along a section of a circle with fixed radius. All of them are facing this object. Originally, this metaphor was created for three-dimensional input devices. In our implementation, these elements are set around the user and he selects them using a ray, with origin in the position of camera and direction of its forward vector.

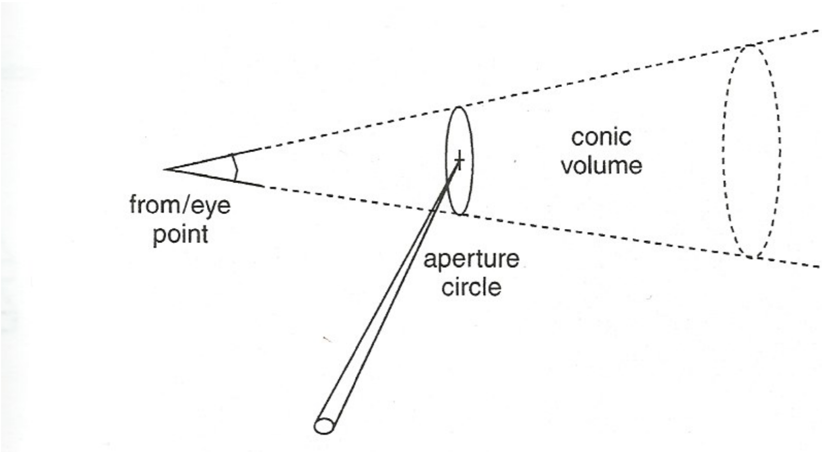


Fig. 2. Aperture selection technique. Aperture circle can be controlled by the user to change its spread of selection. Source: [14]

3 BACKWARD ROTATIONAL VELOCITY PARAMETER IDEA

The idea behind backward rotational velocity parameter is really simple: add additional rotational velocity to implemented ring menu, which lets it rotate around the user but in the opposite direction. Thanks to this, user can rotate his head with smaller angle, because the last elements of menu will appear faster, unlike if it stays in its position all the time.

Equation 3 presents mathematical implementation of backward rotational velocity idea. ω_m is the rotational velocity of our menu. It is important to say, that angle θ is referencing to user's device.

$$\omega_m = -\frac{\Delta\theta + \mu}{\Delta t}, \tag{3}$$

This parameter and "rotational gain" are expected to differ mainly on the field of user experience. Adding backward rotational velocity parameter can make much smoother effect of acceleration than used reorientation technique because we are not manipulating with device rotation, but with objects in the virtual world instead.

Furthermore, user gets much better feedback, while modifying value of this parameter, because it doesn't shift device orientation like "rotational gain" solution. User doesn't need to look at certain point to eliminate visual offset of camera, because he changes in-world object rotation attribute.

4 TEST SYSTEM

Main feature of our test system is the possibility of changing two parameters: ρ and μ, and using them to set a pin number on a ring menu. ρ parameter is a "rotational gain" parameter described in introduction, μ – stands for backward rotational velocity of menu.

According to [11] ring menu metaphor gives the best time results in comparison to other tested metaphors. It is easy to remember positions of items using only one dimension, easy to use and implement [12] and gives opportunity for further improvements which were shown in [8]. That is why we have decided to use ring menu is our test system and evaluation.

Figure 3 shows screenshots taken from the system. On the upper one task has already begun, menu is active, and above it is an actual pin number. On left bottom side, we have the same angle of device but ρ parameter is set to 1.25. On right bottom side we have also the same angle of device but μ equals 0.05.

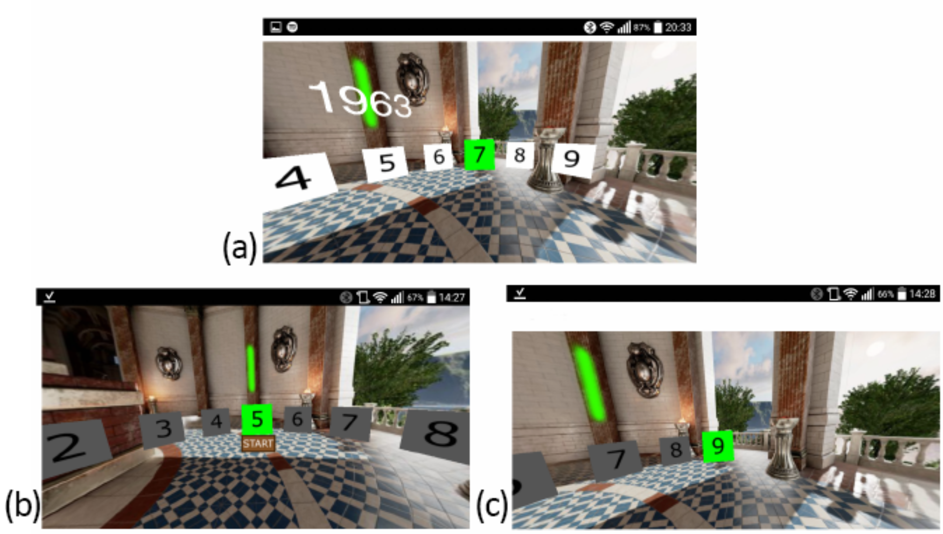


Fig. 3. a) Active test task. The menu contains of nine elements with element "7" selected by the user. Current pin code is displayed above the menu. Parameters values are not displayed to the user to verify that the corresponding values are set in subsequent approaches. b) User is setting ρ parameter using controller (current parameter value is 1.25). Menu elements are inactive – only view change is allowed. c) User is setting μ parameter using controller (current parameter value is 0.05). Menu elements are inactive – only view change is allowed

All data gathered during tasks were being sent to local mySQL database after every task.

Test system was made directly for mobile devices with Android operating system, additionally using VR goggles like Google Cardboard or Google Daydream. The main purpose of our decision was to make the solution universal without relying on products from main VR manufacturers and developers (Oculus, HTC).

Another step to make test system more accessible was the usage of WebGL. Authors decided to use Three.js framework because it's really popular, very customizable and enables creating not only game-like systems.

5 Evaluation

Main purpose of testing sessions was to check whether using one of mentioned parameters or both simultaneously will have an impact on user task completion time. If positive, then for what values of parameters test system offers the best experience. For this purpose HCI evaluation [13] and along with usability and UX tests were conducted [3], [4], [15].

Testing scenario consisted of four stages during which user was asked to enter correctly four, four-digit codes as fast as possible. Codes were the same for each stage allowing time completion task comparison and their were as follows: 1963, 5284, 9347, 2615. Each stage consist of the same steps:

1) setting parameter or parameters (omitted in stage 1),
2) entering codes,
3) repeating steps 1 and 2 a few times,
4) rating a comfort and ease of task completion.

Finally users where asked to pick a stage that was the best in their opinion.

Evaluation began with familiarizing the user with test system and its controls. After that users where asked to enter codes using default setting and repeat this a few times to get average task completion time. During this stage (stage 1), user didn't have to set any parameter because we wanted to gather initial tasks completion time for further analysis.

Second stage enabled changing ρ parameter but only while menu was deactivated. After finishing each task, parameter has been reset, and user had to set its value again, according to his experience with the task before. It is worth mentioning that user didn't know the exact value of both parameters. Thanks to this, authors had a chance to check if user was setting the approximate value on purpose.

Further stages were similar, only with different parameters setting. Third stage enables setting only the μ parameter while fourth gives possibility to test both parameters simultaneously.

During experiment we gathered following data:

- task completion time,
- ρ parameter values set by the user (in range from 1 to 2 and step 0.02),
- μ parameter values set by the user (in range from 0.01 to 0.1 and step 0.01),
- task success (pin validation result along with observed errors).

In conclusion, test scenario was as following:

1) Familiarize users with the test system.
2) Stage 1: entering codes with default settings (without parameters).
3) Stage 2: entering codes after setting ρ parameter (rotational gain).
4) Stage 3: entering codes after setting μ parameter (backward rotational velocity).
5) Stage 4: entering codes after setting both: μ and ρ parameters.
6) User satisfaction ratings for whole test.

For testing purposes we used LG L90 smartphone with Google Daydream goggles. As input device we used SteelSeries Free Mobile Wireless Controller connected with smartphone via bluetooth.

Figure 4 shows a user, taking part of the experiment. Author could easily watch the results thanks to the database website visible on the screen.

Fig. 4. Test participant during experiment

6 RESULTS

Testing group contained nine persons aged 22-25 with good and very good familiarity with virtual reality systems. For further analysis were taken only results with correct task completion. Table 1 presents how many tasks were finished correctly (in percent).

Most of the time, tasks were made correct. Failures in stage one, where users haven't change any parameter, can emerge from former unfamiliarity with the test system. Stage three, which included setting the μ parameter, generated most of the errors according to stages that require modifying parameters, probably because of certain level of unease, which can be seen on questionnaire ratings analysis.

TABLE 1
Task attempts summary with percentage presentation of passed attempts and number of errors for each stage of the evaluation

	Stage 1	**Stage 2**	**Stage 3**	**Stage 4**
Correct attempts	88,89%	100,00%	91,67%	100,00%
Number of errors	4	0	3	0

Figure 5 shows cumulated gain/loss ratio of task completion time for stage two (changing ρ parameter), three (changing μ parameter) and four (changing both parameters) relative to the results from initial stage one. The bars show that usage of additional parameters gives certain amount of speed gain very often.

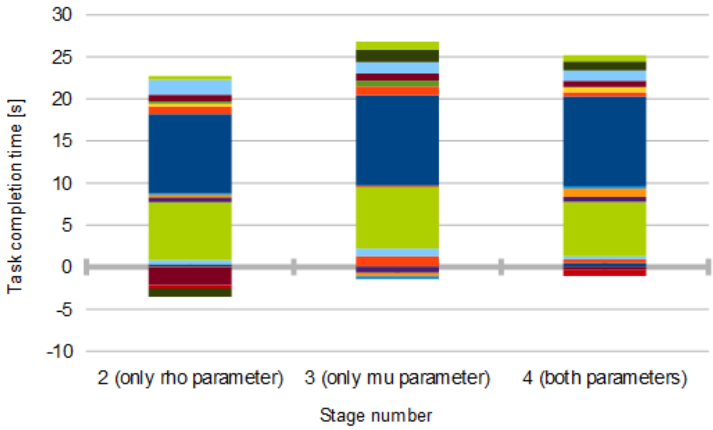


Fig. 5. Gain/loss ratio of task completion time for tasks when users made use of ρ parameter (stage 2), μ parameter (stage 3), and both (stage 4)

Table 2 shows percentage time gain for each stage containing average value, standard deviation, median and confidence interval, relative to stage one results. For each stage the range of time gain isn't very broad which is visible by looking at standard deviation values. During the fourth stage, where users were setting both parameters ρ and μ we have the smallest range in time difference, which means that values of difference in times between this stage and initial stage (without parameters) are closest to its average.

TABLE 2
Percentage time gain relative to completion time results from stage one

	Stage two	**Stage three**	**Stage four**
Average	13,74%	15,85%	20,65%
Stand. deviation	34,61%	29,53%	25,89%
Median	15,69%	22,56%	22,80%
Confidence interval	12,60%	10,75%	9,42%

By taking into consideration confidence interval from Figure 6, we can be 95% sure that for stage two (using only ρ parameter) speed gain will be minimum 1% of completion time. For stage three (using only μ parameter) – minimum 5%, and for stage four (using both ρ and μ parameters) – minimum 11%. Additional t test [3] shows that, for stage two, there is a 2.3% chance that this difference is not significant ($p < 0.05$). For stage three – 1.3% (also $p < 0.05$) and for stage four – 0.3% ($p < 0.01$).

Tests have shown that most commonly used parameters values were as follows:

- between 1,06 and 1,12 using only ρ parameter,
- between 0,02 and 0,03 using only μ parameter,
- between 1,04 and 1,14 for ρ and 0,03 for μ using both parameters.

Therefore, given values can be used as defaults or starting points for individual systems.

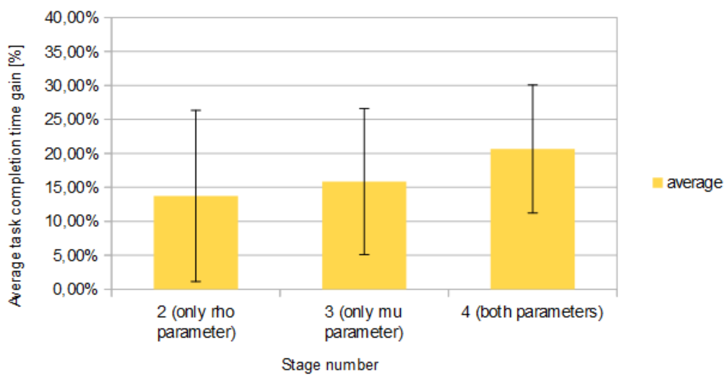


Fig. 6. Average task completion time gain in comparison to first test stage (one without parameters). For analysed group including confidence intervals equal to 0.05. Because of intervals crossing, it is impossible to point the one case, for which time results are the best. It is possible however to show how big the gain will be for each stage

After every stage, users were asked two questions:

- How easy to use was the menu?
- How comfortable the selecting elements using menu was?

Answers were gathered using embedded questionnaire shown in the Figure 7.

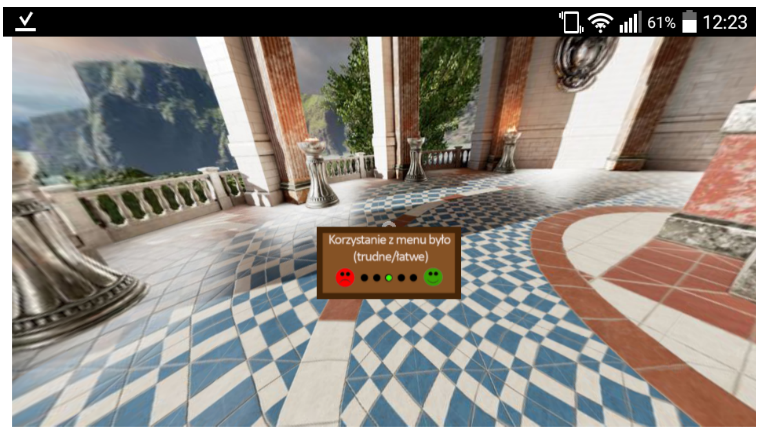


Fig. 7. Embedded questionnaire that collects user impressions after every stage. In scale from one to five, actual value equals 3 and means user's neutral attitude to asked question: "Was using menu hard/easy?"

Figures 8 and 9 present cumulated columnar chart of post-stage ratings given by users.

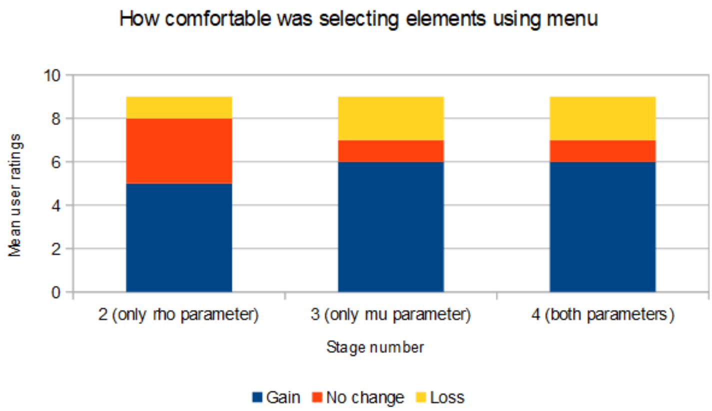


Fig. 8. Mean comfort ratings chart for stages two, three and four relatively to opinions after first stage. It is shown how many positive, neutral and negative impressions had users after each stage

In case of comfort, most of the users had positive opinion (above 50% for each stage). However in stage two (with changing only ρ parameter) 33,33% had neutral attitude to level of comfort.

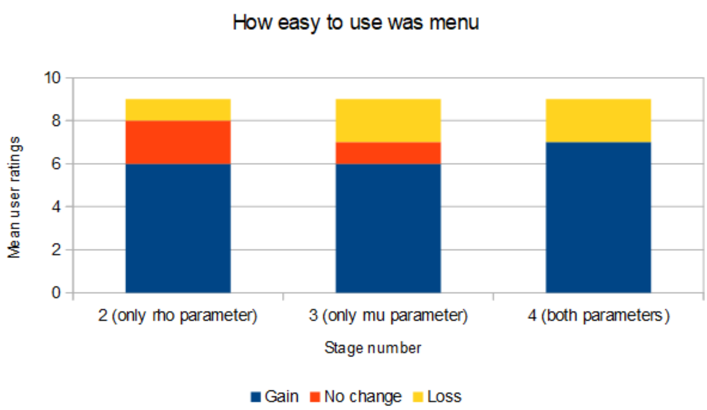


Fig. 9. Mean ease ratings chart for stages two, three and four relatively to opinions after first stage. It is shown how many positive, neutral and negative impressions had users after each stage

Similar situation can be seen on ease rating chart – most of the opinions were positive.

To sum up, according to results shown in Figures 8 and 9, users were mostly satisfied by level of ease and comfort. Most extreme opinions concerned the combination of both parameters ρ and μ. However the answers after tests show that the best option for most users was precisely this combination. 66% of users were the most satisfied after final stage and 33% of them couldn't unambiguously decide between μ parameter and $\rho + \mu$. All results are relative to ratings given after stage one.

Efficiency of our solution is presented as number of operated menu elements per minute and is shown in the Figure 10.

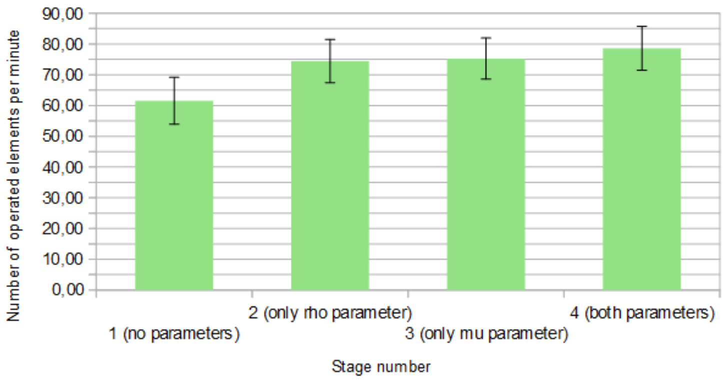


Fig. 10. Efficiency measured using number of operated elements per minute. Confidence intervals have value of 0.95

Once again we used confidence intervals with value of 0.95 and failed attempts weren't taken into consideration. Because only for fourth stage, confidence interval is not crossing with the one in first stage, we can be 95% sure that number of operated elements will be higher than in initial stage for circa 2 elements.

7 Conclusions

Results have shown that adding ρ and μ parameters along with their combination can have positive impact on menu navigation in virtual reality. Still, further validation is needed to show which parameter gives the best results. Also in the third case (combination of both parameters), there is indeed a necessity to set both parameters in appropriate balance. Good opinion given by testers shows that this solution can be comfortable, specially for the users accustomed VR systems.

This model leaves some space for innovations or additional tests. Most interesting way for further idea development is to design another tasks which users can perform. Another possibility lies in implementing the dynamic gain method presented in [2] which showed promising results during pilot studies. Other thing is to test the solution on most popular commercial VR goggles like Oculus Rift or HTC Vive. This technology can create some better opportunities for optimization or implemented interactions.

References

[1] Paludan, A.; Elbeak, J.; Mortensen, M.; Zobbe, M.; Nilsson, N.C.; Nordahl, R.; Reng, L.; Serafin, S.: *Disguising Rotational Gain for Redirected Walking in Virtual Reality: Effect of Visual Density*. 2016.

[2] Hong, S.; Kim, G.J.: *Accelerated Viewpoint Panning with Rotational Gain in 360 Degree Videos* . 2016.

[3] Tullis, T.; Albert, B.: *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Elsevier, Inc. 2013.

[4] Sauro, J.; Lewis, J.R.: *Quantifying the User Experience: Practical Statistics for User Research*. Elsevier, Inc. 2012.
[5] Neth, C.T.; Souman, J.L.; Engel, D.; Kloos, U.; Buelthoff, H.H.; Mohler, B.J.: *Velocity-Dependent Dynamic Curvature Gain for Redirected Walking*. 2011.
[6] Peck, T.C.; Whitton, M.C.; Fuchs, H.: *Evaluation of Reorientation Techniques for Walking in Large Virtual Environments*. 2008.
[7] Razzaque, S.: *Redirected Walking*. 2005.
[8] Gerber, D.; Bechmann, D.: *The Spin Menu: a Menu System for Virtual Environments*. 2005.
[9] Nitzsche, N.; Hanebeck, U.D.; Schmidt, G.: *Motion Compression for Telepresent Walking in Large-Scale Remote Environments*. 2004.
[10] Bowman, D.A.; Kruijff E.; LaViola, Jr.; J.J.; Poupyrev I.: *3D User Interface: Theory and Practice*. 2004.
[11] Gerber, D.; Bechmann, D.: *Design and evaluation of the ring menu in virtual environments*. 2004.
[12] Lian, J.; Green, M.: *JDCAD: A highly interactive 3D modeling system*. 1994.
[13] Lazar, J. et al.: *Research Methods in Human-Computer Interaction*. Wiley Publishing, 2010.
[14] Forsberg, A.; Herndon, K.; Zeleznik, R.: *Aperture Based Selection for Immersive Virtual Environments*. 1996.
[15] Nielsen, J.: *Usability Engineering*. Morgan Kaufmann Publishers Inc.; 1993.
[16] Wojciechowski A.: *3D Scene Exploration with External Controllers*. Towards Modern Collaborative Knowledge Sharing Systems, Studies in Computational Intelligence, Springer Berlin\Heidelberg, pp. 61-70, 2012. Doi: 10.1007/978-3-642-27446-6_6.

Jarosław Andrzejczak Assistant Professor at the Institute of Information Technology, Faculty of Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology. In 2015 he received PhD degree in computer science for interactive information visualization for digital data sets search results. His research interests encompass: usability testing and engineering, User Experience, data and information visualization, user interface design (including game interface design) as well as application of information visualization in UI design.

Jan Bełczącki Graduate with master's degree at Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology on specialization Games and Interactive Systems Technologies. Employed as a programmer (not only VR) in Andrew Lucas Studios.

Lighting models in computer games

Michał Gawron, Urszula Boryczka

University of Silesia, Sosnowiec, Poland
michalgawron.mg@gmail.com, urszula.boryczka@us.edu.pl

Abstract

Lighting models describe the way light reacts in contact with the surface. The aforementioned applies to two types of reflections that together constitute the lighting equation. One of the mentioned reflections is called a diffuse reflection which is responsible for how the objects are lit on the scene. The other is a specular reflection that defines glossiness on the surface of the objects. The goal of the paper is to present lighting models with their approximations and optimizations that increase their efficiency. Numerous techniques are known, but the final decision depends on what is needed at that particular moment.

Index Terms

computer games, computer graphics, light, lighting models, shader

1 Introduction

Computer graphics are being developed since the first computers were introduced. Nowadays, there is a great deal of methods of creating and displaying photorealistic graphics for movies or simulations. However, in most cases we have a previously rendered image that does not require additional real-time calculations. On the opposite side, there are computer games where the dynamic and rich world needs to be processed and displayed in the real-time. Computer games graphics are an element that consumes a significant part of computing power of the computer. Therefore, the functions responsible for image processing, including the lighting models must have a simple form.

Numerous lighting models are known, but the final decision depends on what is needed at that particular moment. The model which has a simple formula and at the same time assures good quality is often chosen for performance reasons. The purpose of the paper is to present lighting models with their approximations and optimizations that increase their performance and maintain quality at the same time.

2 World observation

To better understand the pursuit of photorealism in computer games, one must observe the real world around us, the world that constitutes something natural to us. Frequently, one does not pay attention to how the world functions because one is dealing with it all the time. One of the components that gets ignored is the way the light interacts with the surface. Making observations and drawing the right conclusions brings one closer to creating the correct methods of representing the real world in computer games [11].

2.1 The material classification

Knowing the type of material or surface is very important for the process of creating graphic resources and also for illuminating it later in the game engine. Classification of materials (of surface) can be presented as follows [4] [6]:

1) Electrical conductivity:
   - Conductors (metals).
   - Isolators (non-metals).
2) Reflector:
   - Matt (rough).
   - Shiny (smooth).
3) Directional dependency:
   - Isotropic.
   - Anisotropic.

Thanks to the division presented above one can project realistically any known material or surface.

2.2 Fresnel effect

In computer graphics the word Fresnel refers to differing reflectivity that occurs at different angles [23]. Light that lands on a surface at a grazing angle will be much more likely to reflect than that which hits a surface in the middle of the object. This means that objects will appear to have brighter reflections near the edges. This applies to any type of material, regardless of its characteristics. In theory, each surface can act as a perfect mirror if it is smooth enough and viewed at the right angle. This phenomenon is particularly noticeable for sufficiently smooth and well reflecting surfaces.

2.3 Longitudinal reflections

Another feature observed in real world is the shape of specular reflections. When reflections are observed at a small angle, their shapes are not very different from what the light sources look like. But when the angle starts to expand, the reflections become noticeably longer and stretched towards the observation direction. The most beneficial opportunity to observe the reflections constitute rainy nights, but clear reflections can be also seen on dry surfaces during the day.

2.4 Subsurface scattering

The final, essential observation is the diffusion of light in the internal structure of the surface. The aforementioned causes the opaque items to exhibit the characteristics of the transparent material when observed under appropriate conditions. At the moment of contact of the light rays with the surface, some of them are reflected and some are absorbed and usually brought to the other side of the object. The phenomenon of subsurface scattering is primarily the characteristic of thin materials such as leaves or paper but occurs also in wax, milk or human skin [8].

3 THEORETICAL BASIS

The main problem discussed in this article is the behavior of light in contact with any surface. To better understand the motivation of continuous development of rendering methods, one must ask what light exactly is. According to the dictionary's definition, the visible light (commonly referred to as the light) is an electromagnetic wave with wavelengths in the range of 390 nm to 780 nm. The most important feature of light is its dual meaning [4] – on the one hand it is considered a wave (e.g. law of reflection or phenomena of interference and diffraction). On the other hand, it is perceived in quantum terms, which more or less means that light also has the properties of a stream of particles that carry energy. The attempts to describe the behavior of light in computer games in accordance with the principles of physical and quantum optics would have a very negative effect on performance. Therefore, the most commonly used description of light is the geometrical optics that treats light as a beam of rays.

3.1 Geometrical optics

The assumption of geometrical optics is that light propagates as a stream of rays that run in straight lines from the light source to the point where they encounter an obstacle or a change of a medium. One must keep in mind that the concept of a light ray is not accurate and, with closer examination, it turns out that it is a bit dull with reality. The geometrical optics can be used only in situations where the objects on the light rays path are relatively large. However, in many typical well-known situations in everyday life, the geometric model works quite well – such devices as cameras, glasses, binoculars and telescopes work well on its basis.

3.2 Law of reflection

If the reflecting surface is very smooth, the reflection of light that occurs is called specular or diffuse reflection. The laws of reflection are as follows (Fig. 1):

- The incident ray Θ_i, the reflected ray Θ_r and the normal $\vec{N}$ to the reflection surface at the point of the incidence lie in the same plane.
- The angle α which the incident ray makes with the normal is equal to the angle β which the reflected ray makes to the same normal.
- The reflected ray and the incident ray are on the opposite sides of the normal.

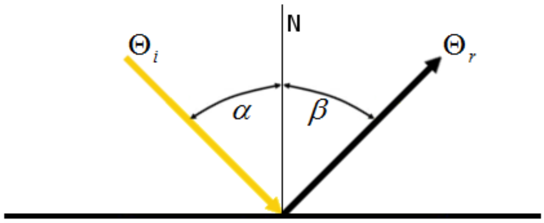


Fig. 1. Graphical representation of the law of light reflection

3.3 Lambert's cosine law

Lambert's cosine law [12] claims that the brightness of the observed surface is directly proportional to the cosine of the angle θ between the direction of the incident of light rays and the normal vector. When the angle θ relative to the surface normal rises, the brightness of the illuminated surface decreases, and vice versa. The reflection described by the law of cosine is known as the Lambert reflection. Surface brightness is expressed by the formula:

$$I = I_L \; cos\theta = I_L \left(\vec{N} \cdot \vec{L} \right) \tag{1}$$

where I_L is intensity of the light source, $\vec{N}$ is surface normal and $\vec{L}$ is direction of the incident ray.

3.4 Bidirectional reflectance distribution function

A bidirectional reflectance distribution function (BRDF) describes the reflectance properties of a surface by specifying the amount of radiance incident from one direction that is reflected into another direction, with respect to the surface normal (Fig. 2). The main characteristics of a physically plausible BRDF are the following:

- The symmetry between incident and reflected directions (Helmholtz reciprocity).
- There is less of total reflected power for a given direction of incident radiation or it is equal to the energy of the incident light (energy conservation).

The BRDF was first defined by Fred Nicodemus in 1970 [14]. The definition is as follows:

$$\begin{aligned} f_r \left(P, \Theta_i \leftrightarrow \Theta_r \right) &= \frac{dL \left(P \rightarrow \Theta_r \right)}{dE \left(P \leftarrow \Theta_i \right)} \\ &= \frac{dL \left(P \rightarrow \Theta_r \right)}{L \left(P \leftarrow \Theta_i \right) cos\theta_i d\omega_{\Theta_i}} \end{aligned} \tag{2}$$

where Θ_i is incoming light direction, Θ_r is outgoing direction, θ_i is angle between the direction of the incident light and the surface normal, dL is radiance and dE is irradiance.

4 Lighting models

The lighting (illumination) model is responsible for approximation of any reflection phenomena, such as diffuse and specular reflection or Fresnel effect. Every illumination model should be described with a simple formula, in such way so that it requires as little resources as possible during rendering. The final form of light reflection is influenced by many parameters describing the surface. For example, it could be the roughness or the ability to conduct electricity. Two very similar surfaces can reflect light in a very different way [11].

Modern shading models are often referred to as physically based that introduce much more complicated calculations. For this reason, some of the discussed lighting models will be presented as the BRDF function. To facilitate the understanding of

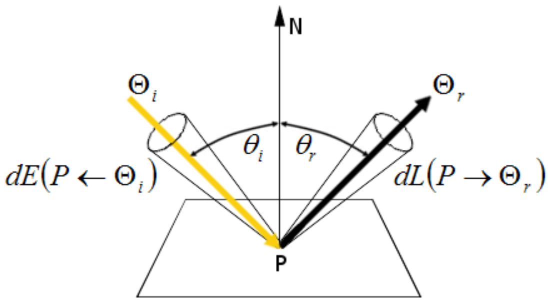


Fig. 2. Diagram showing vectors used to define the BRDF

equations of illumination models, a common diagram of light reflection with marked vectors and angles will be introduced (Fig. 3) where $\vec{N}$ is surface normal, $\vec{L}$ is light direction, $\vec{R}$ is reflected vector, $\vec{V}$ is view direction and $\vec{H}$ is half-vector (sometimes referred to as the microsurface normal).

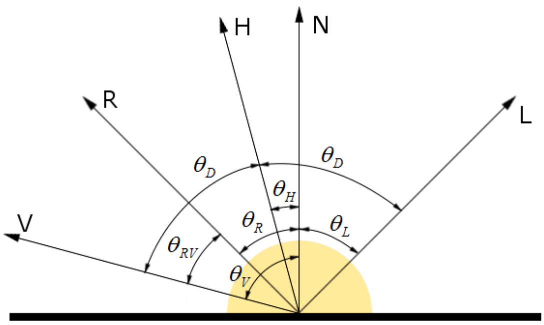


Fig. 3. Reflection geometry used in the description of lighting models

4.1 Diffuse reflection

Diffuse reflection [5] is mainly responsible for how the observed object is lit. If the object is illuminated by a point light source, of which the rays divide uniformly in all directions from one point, the brightness of the illuminated surface changes according to the direction and distance to the light.

4.1.1 Lambert's model

The diffusion model, which is based on cosine law, is called the Lambert's model. It is a basic illumination model that describes the behavior of light reflected on matt surfaces without the gloss, e.g. on chalk. Such surfaces appear to be equally bright from all points of observation.

$$f_d = \frac{1}{\pi} \tag{3}$$

Note that this equation is devoid of $cos\,\theta_L$ because this factor is contained directly in the BRDF equation 2.

4.1.2 Minnaert model

The solution proposed by Minnaert [13] is a modification of the Lambert's model where the amount of reflected light from the surface depends on the direction of observation. For this purpose, the Fresnel effect controlled by a separate parameter allows to darken the edges of the object. This gives the material the appearance of velvet or satin,

$$f_d = cos\theta_L \left(cos\theta_L \, cos\theta_V\right)^n \tag{4}$$

where n is value of the edge darkening.

4.1.3 Oren-Nayar model

The Oren-Nayar [15] model is used to simulate the diffusion of light on matte surfaces, such as concrete or sand for which the Lambert model is insufficient. The Oren-Nayar model is characterized by two features: it depends on the observation angle and uses the roughness coefficient of the surface. For calculation of light value on rough surfaces, the Oren-Nayar model uses the microfacet model proposed by Torrance and Sparrow [19] which assumes that the surface consists of symmetrical longitudinal V-shaped cavity:

$$f_d = \frac{1}{\pi} cos\theta_L \left(A + BC \, sin\rho \, tan\beta\right) \tag{5}$$

$$\beta = min\left(\theta_V, \theta_L\right) \tag{6}$$

$$\rho = max\left(\theta_V, \theta_L\right) \tag{7}$$

$$A = 1 - 0.5 \frac{\alpha^2}{\alpha^2 + 0.33} \tag{8}$$

$$B = 0.45 \frac{\alpha^2}{\alpha^2 + 0.09} \tag{9}$$

$$C = max\left(0, cos\left(\theta_V - \theta_L\right)\right) \tag{10}$$

where α is the roughness of the surface.

4.1.4 Burley's model

The Burley's empirical model [2] is an innovative alternative to the Lambert's model. It eliminates the dark edges effect and is dependent on the roughness of the material and the angle of observation. The Burley's model equation uses the Fresnel effect, thanks to which the reflection of the light at the edge angles for smooth surfaces is reduced by half. But for rough surfaces the brightness of the reflection increases by 2.5 times:

$$f_d = \frac{1}{\pi} \left(1 + \left(F_{D90} - 1\right)\left(1 - cos\theta_L\right)^5\right) \left(1 + \left(F_{D90} - 1\right)\left(1 - cos\theta_V\right)^5\right) \tag{11}$$

$$F_{D90} = \frac{1}{2} + 2\, cos^2\theta_D \, \alpha \tag{12}$$

where α is the roughness of the surface.

4.2 Specular reflection

The previously described diffusion models are only the first step, describing how the light is reflected. The expansion of the idea of diffuse lighting is the addition in the equation of the amount of specular reflection [5] that can be observed on every shiny surface. With these two types of reflection, the lighting equation is complete.

4.2.1 Phong model

The Phong model [16] is based on two assumptions: the specular reflection maximum occurs at an angle α equal to zero and rapidly decreases with its increase. This rapid decrease is approximated by $cos^n(\alpha)$, while the exponent n characterizes the behavior of specular reflection for a given material. The normalized BRDF version of the Phong model has the form:

$$f_s = \frac{n+2}{2\pi} cos^n \theta_{RV} \tag{13}$$

where n is value of the surface gloss.

4.2.2 Blinn-Phong model

The Blinn-Phong (or Blinn) reflection model [1] is a modification of Phong's model developed by James Blinn. This model ensures that the specular reflection is longitudinal and resembles a reflection in the real world:

$$f_s = \frac{n+2}{4\pi\left(2 - 2^{-\frac{n}{2}}\right)} cos^n \theta_H \tag{14}$$

where n is value of the surface gloss.

4.2.3 Cook-Torrance model

It is the first lighting model based on the physical properties of the surface created by Robert Cook and Kenneth Torrance in 1981 [3]. This model is much closer to reality than previously described models. The solution proposed by Cook and Torrance introduces the dependence of the brightness of reflection from the observation angle, and therefore the surface reflects more light in some directions than others. This model also uses the microfacet model developed by Torrance and Sparrow [19]. The general form of the model is described by equation 15 where F is the Fresnel coefficient. G defines the geometrical attenuation factor for the microfacet model. Factor D describes the distribution of microfacet inclinations and thus determines the surface roughness. In consequence:

$$f_s = \frac{F_{CT}}{\pi} \frac{G_{CT}\, D_{CT}}{cos\theta_L\, cos\theta_V} \tag{15}$$

$$F_{CT} = \frac{1}{2} \frac{(g-c)^2}{(g+c)^2} \left(1 + \frac{(c\,(g+c)-1)^2}{(c\,(g-c)+1)^2}\right) \tag{16}$$

$$c = cos\theta_D \tag{17}$$

$$g = \sqrt{\eta^2 + c^2 - 1} \tag{18}$$

$$F_0 = \left(\frac{\eta - 1}{\eta + 1}\right)^2 \tag{19}$$

$$\eta = \frac{1 + \sqrt{F_0}}{1 - \sqrt{F_0}} \tag{20}$$

$$G_{CT} = min\left(1, \frac{2\, cos\theta_H\, cos\theta_V}{cos\theta_D}, \frac{2\, cos\theta_H\, cos\theta_L}{cos\theta_D}\right) \tag{21}$$

$$D_{CT} = \frac{1}{\pi\, \alpha^2\, cos^4\theta_H}\; e^{-\left(\frac{tan\,\theta_H}{\alpha}\right)^2} \tag{22}$$

where α is the roughness of the surface and η is the refractive index of the material.

4.2.4 Ward model

Ward has created an illumination model compatible with both types of surface: isotropic and anisotropic [22]. The Ward model uses the Gaussian distribution as a roughness coefficient. Unlike the Cook-Torrance model, it does not use the Fresnel and the geometric attenuation factors which require additional computing effort. In their place, single normalization factor is introduced:

$$f_s = \frac{1}{\sqrt{cos\theta_L\, cos\theta_V}}\; \frac{e^{-tan^2\theta_H\left(\frac{cos^2\phi}{{\alpha_x}^2} + \frac{sin^2\phi}{{\alpha_y}^2}\right)}}{4\pi\, \alpha_x\, \alpha_y} \tag{23}$$

where α_x is the standard deviation of the surface slope in the X direction of the surface plane, α_y is the standard deviation of the surface slope in the Y direction, ϕ is the azimuth angle of the half vector projected into the surface plane.

4.2.5 GGX model

GGX model [20] is based on the Cook-Torrance model, but has a better function of reflectance distribution. Thanks to the aforementioned, it is much closer to reality and one can easier obtain specificity on the observed surface (Fig. 4). The GGX model is available in isotropic and anisotropic versions. It is today's most commonly used specular reflection model. In the following equations the presented modifications are described:

$$f_s = \frac{F_{CT}\, G_{SmithGGX}\, D_{GGX}}{4\, cos\theta_L\, cos\theta_V} \tag{24}$$

$$D_{GGX} = \frac{\alpha^2}{\pi\left(1 + cos^2\theta_H\,(\alpha^2 - 1)\right)^2} \tag{25}$$

$$G_{SmithGGX} = \left(\frac{2}{1 + \sqrt{1 + \alpha^2\, tan^2\theta_L}}\right)\left(\frac{2}{1 + \sqrt{1 + \alpha^2\, tan^2\theta_V}}\right) \tag{26}$$

where α is the roughness of the surface.

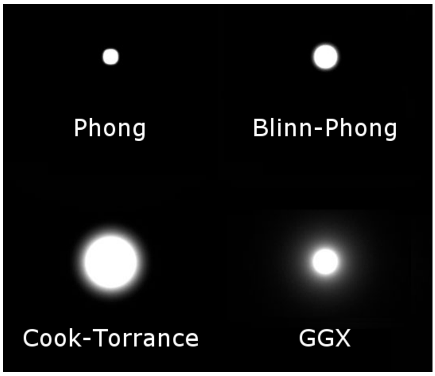


Fig. 4. Comparison of the reflectance distribution of different models

4.2.6 Kajiya-Kay model

Kajiya-Kay model [9] is an anisotropic version of the Phong model for the reflections observed on the hair. In the equation, it replaces the normal vector of the surface with a vector lengthwise one of the axes (Fig. 5). To this day, it is one of the most popular models of anisotropic light reflection off the hair surface. Its popularity occurs due to the simplicity and its large possibilities in regards of the enhancement of the reflection effect.

$$f_s = (cos\theta_{TR}\, cos\theta_{TV} + sin\theta_{TR}\, sin\theta_{TV})^n \tag{27}$$

where n is value of the surface gloss.

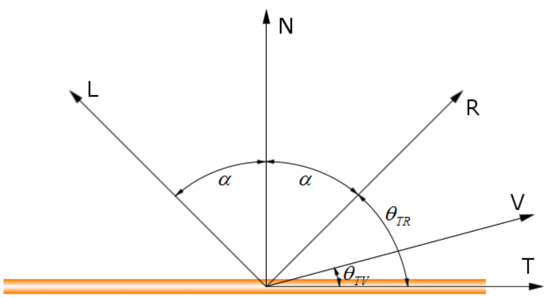


Fig. 5. The geometry of the reflection of the Kajiya-Kay model on hair surface

5 FUNCTIONS OPTIMIZATION

Some functions describing the behavior of light in contact with the surface are difficult to implement due to the amount of operations that must be performed (Table 1). However, in many cases one can perform approximation. By finding a function that brings close results, one can reduce the amount of shader instructions and increase the performance. In the graphs can be seen that the simplified versions of functions are often very inaccurate. However, this applies only to selected ranges of input data. Usually these values are very rarely used. In other cases, the differences in the appearance of illuminated objects on the scene are not noticeable.

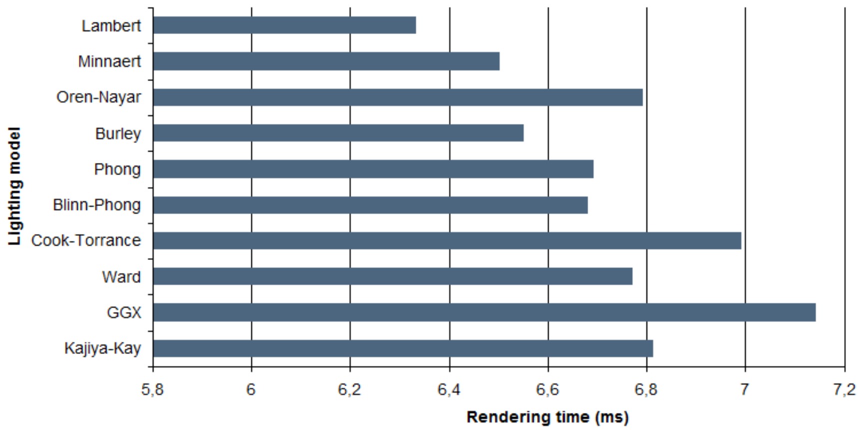


Fig. 6. Performance of lighting models in Unity game engine for DirectX 11 platform

Another factor that improves the performance of illumination models is the optimization of their shader code. When writing shaders, one must avoid certain instructions. These include: trigonometric functions (sine, tangent etc.) or exponential functions. Another important aspect is writing the code in MAD-form, avoid division and separate scalars and vectors.

On the graph on Fig. 6, one can see that the number of shader instructions (Table 1) affects the rendering time of the single frame. More complex lighting models have longer rendering times. The complexity of test scenes was the same for all methods – each scene consisted of 2250 objects and 55 dynamic point lights and used a forward rendering path.

TABLE 1
Number of shader instructions for lighting models in Unity game engine (DirectX 11)

Lighting model	**Shader instructions**
Lambert	15
Minnaert	25
Oren-Nayar	66
Burley	40
Phong	28
Blinn-Phong	34
Cook-Torrance	69
Ward	45
GGX	74
Kajiya-Kay	45

5.1 Approximation of the exponential function

The exponential function $cos^n\theta$ is very common in the equations of illumination models that is written in the shader code as a `pow(cosTheta, n)`. The exponentiation function is usually implemented as:

```
pow(a, b) = exp2(log2(a) * b)
```

which gives three ALU instructions: `LOG2`, `MUL` and `EXP2`. To optimize this function, we used the spherical Gaussian approximation [21] which is written in the form of an exponential function with base e:

$$cos^n\theta = e^{n(cos\theta-1)} \tag{28}$$

By using this approximation, we write the exponential function as:

```
pow(cosTheta, n) = exp(n * (cosTheta - 1)) =
exp2(n * (cosTheta - 1) / log(2)) =
exp2(n / log(2) * (cosTheta - 1))
```

After refactoring of the code and calculating the constant `1 / log(2)`, the function gains the following form:

```
A = n * 1.4427
pow(cosTheta, n) = exp2(A * cosTheta - A)
```

This form of writing also translates into three ALU instructions: `MUL`, `MAD` and `EXP2`. The amount of instructions relative to the original notation of the exponentiation does not change. `LOG2` is replaced by `MUL` instruction. For better optimization we skip the multiplication `A = n * 1.4427` and we simply use:

```
pow(cosTheta, n) = exp2(n * cosTheta - n)
```

Compared to a standard power implementation, we effectively save one instruction. On the graph on Fig. 7, one can see that this approximation is inaccurate. However, these differences can be easily eliminated by the using a higher gloss value by the 2D/3D artists when creating textures: $n' = 1.44n$.

5.2 Approximation of the normalization factor of the Blinn-Phong model

The original form of the normalization factor of the Blinn-Phong reflectance model presented in equation 14 requires a large number of operations. However, this factor can be reduced to a linear function. In the *gnuplot* program, we fitted the normalization factor to the linear function $ax + b$ (29). Thanks to this the simplified calculation of the normalization factor for the Blinn-Phong model requires only one ALU instruction: `MAD`:

$$f_s = 0.0397808n + 0.0823107\, cos^n\theta_H \tag{29}$$

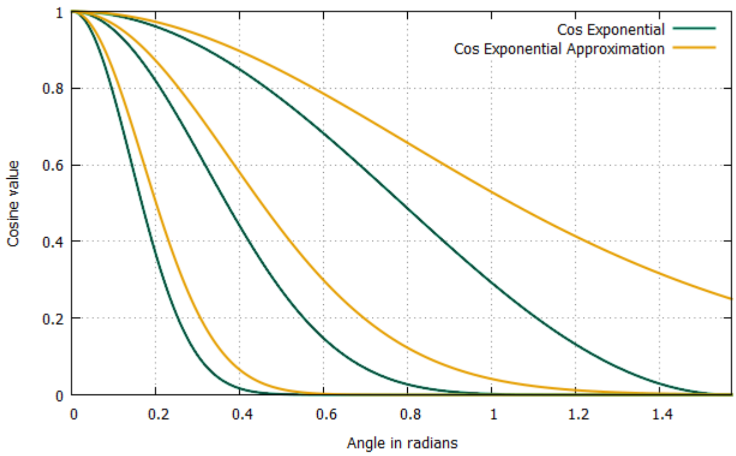


Fig. 7. Approximation of exponential cosine function $cos^n\theta$ for different values for n, from left: 50, 10, 2

On the graph presented on Fig. 8, this approximation for the gloss value $n < 10$ deviates from the curve of the original normalization factor. Considering the fact that gloss values below 10 are applied very rarely, this approximation is admittedly correct.

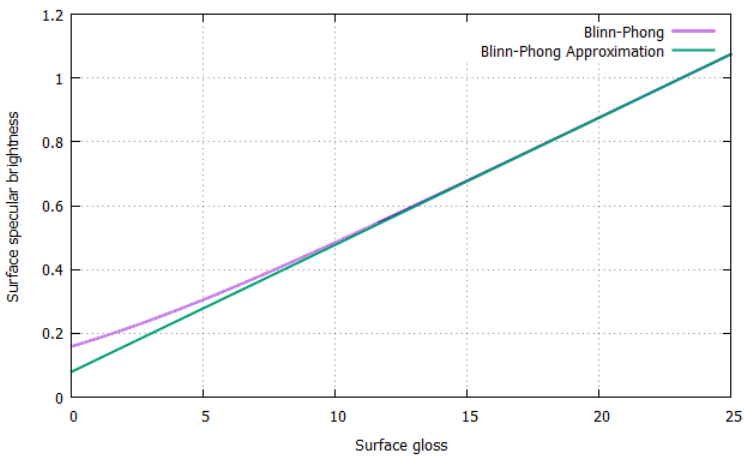


Fig. 8. Approximation of normalization factor of Blinn-Phong model

5.3 Approximation of the Fresnel coefficient

In the era of physical rendering, the Fresnel factor is an inseparable part of modern game engines and rendering applications. However, using the Fresnel equation as proposed by Cook and Torrance requires a large amount of mathematical operations.

The aforementioned was noticed by Christophe Schlick in 1994 [18]. Schlick proposed a much simpler version of the Fresnel equation that works very well for insulators:

$$F_{Schlick} = F_0 + (1 - F_0)(1 - cos\theta_D)^5 \tag{30}$$

Schlick approximation of the Fresnel equation uses the base reflection value F_0 for angle $cos\theta_D$ that equals 0. On the basis of this value, along with the increase of the observation angle, the value of the reflected light increases until the value of 1 is reached. However, for high values of the base reflection that are the characteristic of metals, the Schlick equation starts to deviate from the original form.

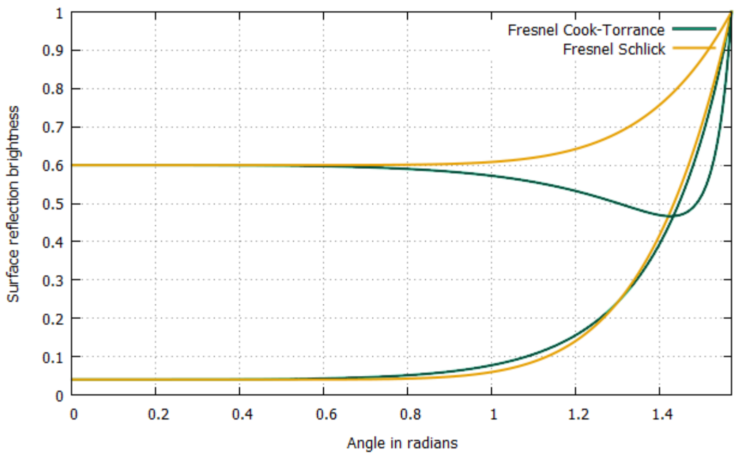


Fig. 9. Schlick Fresnel approximation

On the graph on Fig. 9, the absence of the characteristic drop just before the rapid growth is noticeable. However, such large deviation is acceptable in computer games for one reason. In most cases, the objects on the scene consist of non-conductive materials for which the base reflection value usually falls between 2 and 8%. Pure metals, such as like steel or gold are very rare even in everyday life. Despite this simplicity, the Fresnel-Schlick equation requires the exponentiation. Again we applied the spherical Gaussian approximation [21]:

$$(1 - cos\theta_D)^5 = e^{a\, cos\theta_D} \tag{31}$$

Using the *gnuplot* software we fitted the function and we found the value of parameter $a = -5.81865$, which allows the following to be written the shader code:

```
pow(1 - cosTheta, 5) =
exp(-5.81865 * cosTheta) =
exp2(-5.81865 * cosTheta / log(2)) =
exp2(-8.4 * cosTheta)
```

As a result, only two ALU instructions are executed: `MUL` and `EXP2` which helps to spare twice as much as in case of the original code (`SUB`, `LOG2`, `MUL` and `EXP2`).

5.4 Optimization of the GGX reflectance model

The first step that can be performed during the optimization of the GGX reflectance model is the conversion of Cook-Torrance's Fresnel equation to Schlick's approximation. Performance gains will be noted immediately. The second step is to change the function of the geometrical attenuation factor to a cheaper equivalent. Schlick proposed approximation [18] in a simple formula (equation 32). Schlick's approximation is not accurate but sufficient to obtain reliable and correct results (Fig. 10).

$$G_{SchlickGGX} = \left(\frac{cos\theta_L}{cos\theta_L(1-k)+k}\right)\left(\frac{cos\theta_V}{cos\theta_V(1-k)+k}\right) \tag{32}$$

$$k = \alpha\sqrt{\frac{2}{\pi}} \tag{33}$$

Better conformity of Schlick's approximation can be achieved by changing the writing of parameter k to an even simpler form [10]:

$$k = \frac{\alpha}{2} \tag{34}$$

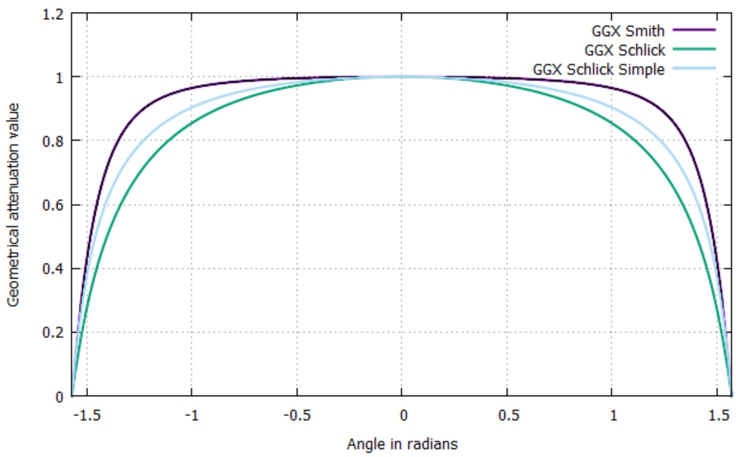


Fig. 10. Approximation of the geometrical attenuation factor of GGX model

The last step in optimizing the GGX reflectance model is the combination of the geometric attenuation function with the denominator of the BRDF equation (24). This combination creates a new visibility term V [7] whose form is as follows:

$$V_{GGX} = \frac{1}{2}\left(\frac{1}{cos\theta_L\,(1-k)+k}\right)\left(\frac{1}{cos\theta_V\,(1-k)+k}\right) \tag{35}$$

Therefore, the BRDF equation of the GGX model will take the following form:

$$f_s = F_{Schlick}D_{GGX}V_{GGX} \tag{36}$$

5.5 Optimization of the Kajiya-Kay specular model

The equation of the reflectance model proposed by Kajiya-Kay contains two sine functions (equation 27). At the time of writing the shader code, every time it is bought forth, it should be converted to a graphics card – friendly code. In addition, one should designate a costly reflected vector. However, the strategy of the Blinn-Phong model can be used and one can bring into play the half-vector within the optimization [17] (Fig. 11). Thanks to this, the light reflection equation of Kajiya-Kay model has only one sine function raised to a power:

$$f_s = sin^n \theta_{TH} = \left(\sqrt{1 - \left(\vec{T} \cdot \vec{H}\right)}\right)^n \tag{37}$$

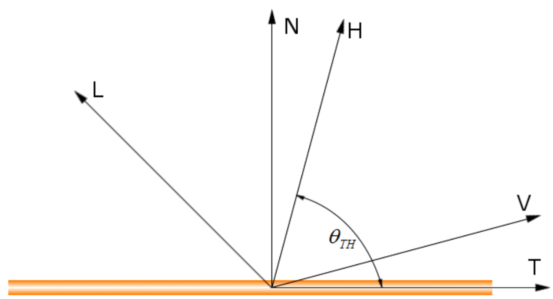


Fig. 11. The geometry of the reflection of the Kajiya-Kay model using the half-vector

6 Conclusions

The choice of diffuse and specular lighting model is a very important step. There are many algorithms that very well represent the behavior of light in computer games. However, they often require a different set of input data. A good example is the surface roughness coefficient which is replaced by the gloss factor for other illumination models. Both parameters refer to the same material property, but are represented differently. The aforementioned directly affects the production process of graphic resources, most of all – of textures.

On the graph on Fig. 12, one can see that the optimizations presented in the article have reduced the number of shader instructions. In some cases it is up to 30% less instructions. The most important thing to note is that the optimizations did not affect the appearance of the illuminated objects (Fig. 13 and 14).

Such high availability of various implementations makes it difficult to determine which one is the best. In some situations, the Lambert diffusion model can be sufficient, but in others it may be necessary to use a more complex solution. The same applies to the specular reflection. If only the sense of shininess of the object is required, it is beneficial to use a simple and fairly close-to-reality Blinn-Phong reflectance model. On the other hand, when physical correctness is important, the current popular GGX model should be used.

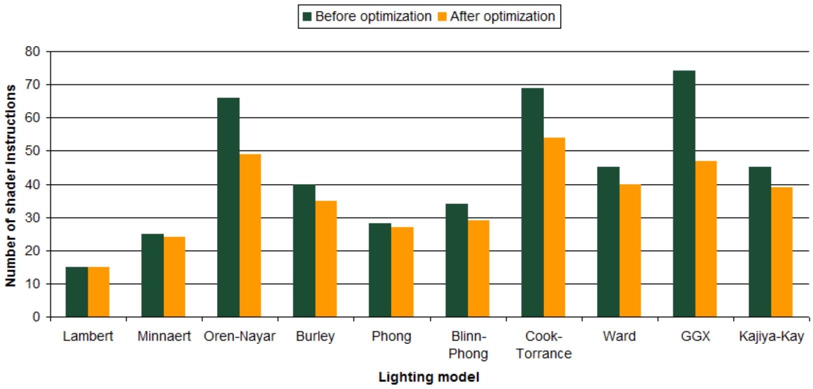


Fig. 12. Cost of implementation of lighting models

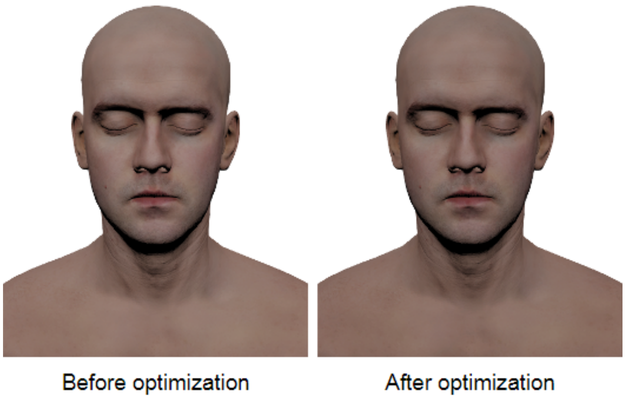


Fig. 13. Comparison of Oren-Nayar model before and after optimization

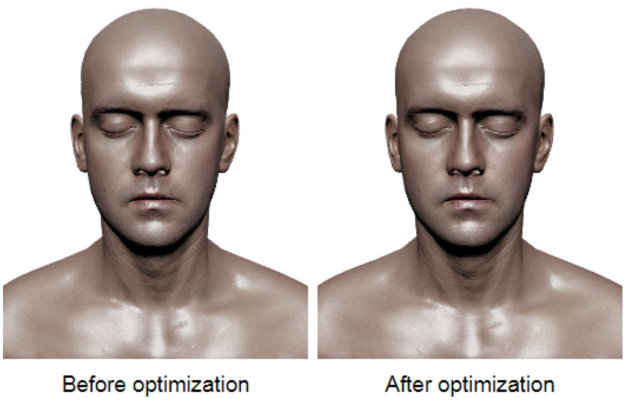


Fig. 14. Comparison of GGX model before and after optimization

REFERENCES

[1] Blinn, J.F.: *Models of Light Reflection for Computer Synthesized Pictures*. SIGGRAPH 77, p. 192-198, 1977.

[2] Burley, B.: *Physically-Based Shading at Disney*. https://disney-animation.s3.amazonaws.com/library/s2012_pbs_disney_brdf_notes_v2.pdf, 20 August 2017.

[3] Cook, R.L.; Torrance, K.E.: *A Reflectance Model for Computer Graphics*. ACM Transactions on Graphics, Volume 1(1), p. 7-24, 1982.

[4] Dempski, K.; Viale, E.: *Advanced Lighting And Materials With Shaders*. Wordware Publishing Inc.; 2005.

[5] Ferguson, R.S.: *Practical Algorithms for 3D Computer Graphics, Second Edition*. A K Peters/CRC Press, 2013.

[6] Fosner, R.: *Real-Time Shader Programming*. Morgan Kaufmann, 2003.

[7] Hoffman, N.: *Background: Physics and Math of Shading*. http://blog.selfshadow.com/publications/s2013-shading-course/hoffman/s2013_pbs_physics_math_notes.pdf, 20 August 2017.

[8] Jimenez, J. et al.: *Separable Subsurface Scattering*. Computer Graphics Forum, Volume 34(6), p. 188-197, 2015.

[9] Kajiya, J.T.; Kay, T.L.: *Rendering fur with three dimensional textures*. ACM SIGGRAPH Computer Graphics, Volume 23(3), p. 271-280, 1989.

[10] Karis, B.: *Real Shading in Unreal Engine 4*. https://cdn2.unrealengine.com/Resources/files/2013SiggraphPresentationsNotes-26915738.pdf, 18 November 2017.

[11] Lagarde, S.; de Rousiers, C.: *Moving Frostbite to Physically Based Rendering 3.0*. https://media.contentapi.ea.com/content/dam/eacom/frostbite/files/course-notes-moving-frostbite-to-pbr-v2.pdf, 18 November 2017.

[12] Luna, F.: *Introduction to 3D Game Programming with DirectX 11*. Mercury Learning & Information, 2012.

[13] Minnaert, M.: *The reciprocity principle in lunar photometry*. Astrophysical Journal, Volume 93, p. 403-410, 1941.

[14] Nicodemus, F.E.: *Reflectance nomenclature and directional reflectance and emissivity*. Applied Optics Volume 9(6), p. 1474-1475, 1970.

[15] Oren, M.; Nayar, S.K.: *Generalization of Lambert's Reflectance Model*. SIGGRAPH 94, p. 239-246, 1994.

[16] Phong, B.T.: *Illumination for Computer Generated Pictures*. Communications of the ACM, Volume 18(6), p. 311-317, 1975.

[17] Scheuermann, T.: *Practical Real-Time Hair Rendering and Shading*. http://amd-dev.wpengine.netdna-cdn.com/wordpress/media/2012/10/Scheuermann_HairSketchSlides.pdf, 18 November 2017.

[18] Schlick, C.: *An Inexpensive BRDF Model for Physically-based Rendering*. Computer Graphics Forum, Volume 13(3), p. 233-246, 1994.

[19] Torrance K.E.; Sparrow E.M.: *Theory for Off-Specular Reflection From Roughened Surfaces*. Optical Society of America, Volume 57(9), p. 1105-1114, 1967.

[20] Walter, B. et al.: *Microfacet models for refraction through rough surfaces*. EGSR'07 Proceedings of the 18th Eurographics conference on Rendering Techniques, p. 195-206, 2007.

[21] Wang, J. et al.: *All-frequency rendering of dynamic, spatially-varying reflectance.*. ACM Transactions on Graphics (TOG) – Proceedings of ACM SIGGRAPH Asia 2009, Volume 28(5), 2009.

[22] Ward, G.J.: *Measuring and modeling anisotropic reflection*. ACM SIGGRAPH Computer Graphics, Volume 26(2), p. 265-272, 1992.

[23] Wloka, M.: *Fresnel Reflection. SDK White Paper*. http://developer.download.nvidia.com/SDK/9.5/Samples/DEMOS/Direct3D9/src/HLSL_FresnelReflection/docs/FresnelReflection.pdf, 18 November 2017.

Michał Gawron, mgr inż. He completed engineering studies in 2015 and he received his Master's degree in 2017 at the University of Silesia in Sosnowiec. Employee of The Farm 51 S.A. in Gliwice as VFX Artist since 2014. He is a lecturer at the Computer Games Programmer specialization in the Division of Algorithmic and Computational Intelligence in Institute of Computer Science since 2015. His current research interests are in the field of computer games: primarily the techniques of reproducing the properties of materials and surfaces in the virtual world of the game.

Urszula Boryczka, dr hab. She completed her Master's degree at University of Silesia in Sosnowiec in 1984 and she received Ph.D. degree at University in Wroclaw in 1993. She holds Doctor's of Science (Habilitation) degree in Computer Science from the Polish Academy of Science in Warsaw in 2009. At present she is a director of Division of Algorithmic and Computational Intelligence in Institute of Computer Science. Her current research interests are in the field of swarm intelligence: ant colony optimization, particle swarm optimization and evolutionary computation. She has published many articles concerning the subject of ant colony optimization and data mining tasks. She has also been a speaker in many international conferences and a member of several program committees of international conferences during the last 3 years.

Biophysical signals in game testing – EEG signal classification

Dominik Szajerman, Adrian Smagur,
Sławomir Opałka, Adam Wojciechowski

Lodz University of Technology, Łódź, Poland
dominik.szajerman@p.lodz.pl

Abstract

Modern technology advancement states that the last step of communication between human and electric devices would be by mental commands. To achieve this, EEG signal analysis improvement is held by wide range of researchers both in communication (BCI control) and mental state analysis (affection feedback) cases. For game industry, this can mean many possibilities of product user experience (UX) testing improvement. In our work, we present a new approach for mental tasks classification using convolutional neural networks (CNN) with adaptively solved parameters to cover both the spatial and noisy EEG characteristics.

Index Terms

BCI, EEG, CNN, Game testing, Affective feedback

1 Introduction

For the interactive media industries like movie or games development companies, biophysical interfaces are to the time being, most attractive as a mean for more accurate feedback from the users during testing stage of product development. Brain-computer interfaces (BCI) rely on recognition of users thoughts as a command messages rather than trying only to detect a discrete physical activity [1]. From many other biophysical signals like magnetoencephalography (MEG), functional magnetic resonance imaging (fMRI) and positron emission tomography (PET), electroencephalogram (EEG) signals are often considered for this task. Being the most convenient to collect from both user and hardware sides the EEG already has a set of proved information paradigms which can be used for mental task recognition like imagination of simple abstract definitions (P300 paradigm) or imagination of left/right hand body movements (12.5-15.5 Hz frequencies). However, EEG vulnerability to noise from outer factors and its still unknown complete characteristic leads to the need of constant improvement in both feature extraction and classification.

2 Related Work

To use proper signal type such as EEG in communication task, two stages: feature extraction [2], [3] and pattern classification should be performed to achieve desired analysis effectiveness of the carried information. The second phase is an application of a discrete classifier such as linear discriminant analysis (LDA) [4] that are hyperplane separation based, support vector machines (SVM) [5]–[7] that are another group of methods for between-classes hyperplane construction. Bayesian classifiers [8] assign a feature vector to every class for better pattern recognition. Other method, Gaussian mixture model, is a clustering method where the data is modeled by a probability density function [9]. Quick development of classification methods allowed to use them on various platforms including less powerful, like mobile devices [10].

Another group of classification methods – artificial neural networks (ANN), are also widely used for this purpose as their effectiveness in pattern recognition is already proved, depending on the subject [14]. Some of them have already been used in EGG classification for BCI usage. The most popular one – Multi-layer perceptron (MLP), tends to result in satisfying results, yet this type of NN also is vulnerable to overtraining because of its simplicity, especially for such uncertain signal as EEG [4]. In context of BCI, neural networks may be enhanced to a proper kind to improve their results e.g. by back-propagation of errors (BPNN), genetic algorithms (GANN), particle swarm optimization (PSONN) and improved particle swarm optimization (IPSONN) [11] or backtracking search optimization algorithm (BSANN) based on evolutionary algorithms [3].

All mentioned NN methods were tested through BCI Competition III, data set V [12] which is a well-known standard collection useful for comparing effectiveness of mental tasks classification. In the year of the test publication, the best solution was not based on NN, but on statistical discrimination with online discrimination improvement [13] , but later developed NN solutions improved this result. Amongst various published solutions there was no CNN-based one. Convolutional Neural Networks are biologically-inspired by visual cortex. This paper proposes CNN approach with adaptive neurons weights update – AdaGrad solver, as the new approach of classifier for BCI applications, that can result in improved effectiveness among other solutions.

3 METHOD

Convolutional neural nerworks architecture base on a simple principle that only some of the outputs are connected to the inputs of the following layer [14]. Because of its recognized effectiveness against image classification tasks, it can be also successfully used on other multidimensional-organized data. EEG frame is recorded as a multi-dimensional cortical areas evoked potential vector, thus CNN have also an implied potential for its analysis.

As dataset [12] was collected by 8 centro-parietal electrodes (C3, Cz, C4, CP1, CP2, P3, Pz, and P4 in standard 10-20 placement system). With sampling rate of 512Hz, the single data record was saved every 62.5 ms (16 times per second) with the power spectral density estimated from raw EEG in the 8-30 Hz band (every 2 Hz – 12 intervals for each electrode). The most predominant number of overall convolutional channels was 96 (8 channels for each electrode times 12 frequency interval components). In advance the slicer input layer divides and provides the signal to Convolutional Layers (CONV_i_j where i – layers index of its type in its channel and j – number of frequency band channel). Doing so, the signal is analyzed by every frequency band channel (96 channels, 12 bands for every 8 electrodes) represented by single vector 16x1. In our assumptions there is no possibility of mixing correlations between those channels like it would be present in a single channel analyzing whole vector 96x16. The multichannel approach in comparison with single channel solution (Fig. 1 left), resulted in improvement of 2-8%. The rate of classification has differed depending on the subject data.

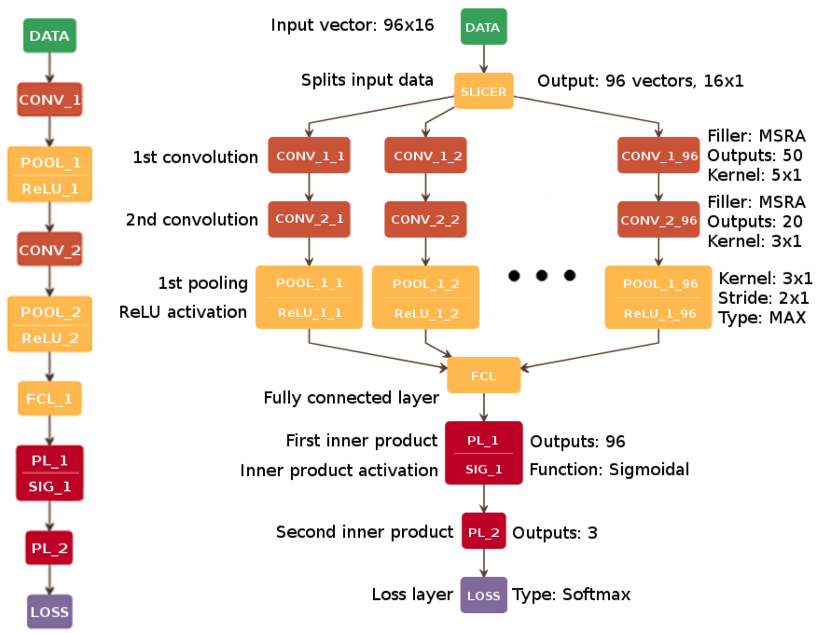


Fig. 1. Early version of the CNN for single channel analysis (left) and proposed CNN architecture (right): DATA – input data, CONV_i(_j) – Convolution Layers, POOL_i(_j) – Pooling Layers, ReLU_i_j – Rectified Linear Unit Activation Layers, FCL_i – Fully Connected Layer, PL_i – Perceptron Layers, SIG_i – Sigmoidal Activation Layer, LOSS – Softmax Loss layer, i – layers index of its type in its channel and j – channel's index

In the final CNN96 (Fig. 1 right), the convolution is carried out by two following CONV layers. The proposed amount of convolutional layer output connections was 50 for every first one and 20 for every second one in the network scheme channel while the filters weights were set with MSRA rectifier which is more suitable for ReLU activation than sigmoid-like functions [15]. After those two convolutions the MAX Pooling Layer (POOL_i) was added, as to reduce the dimension of the network – downsampling data vector by given kernel (filter) and chosen stride dimension. Next the Fully Connected Layer (FCL_i) that connects the results from each previous convolutional layer was used. Subsequent two Perceptron Layers (PL_i), containing 96 in the first layer and 3 outputs for each class for the final result, provide input for accuracy evaluation within Accuracy Layer (AL_i). AL calculates the correct classification of the current data vector and refers it to the accuracy threshold from the learning phase. Finally softmax loss (LOSS) function layer clamps and normalizes the output values.

The main characteristics of CNN are the convolution layers which are similar to typical perceptron layer but their task is to create a matrix of features. The convolution process involves calculating dot product of a spatial region in the input data by so called filters (Fig. 2).

This matrix of features (Feature Map) consists of scalars produced by every filter application to the input data spatial region. While convolution proceeds, the filter is applied to a discrete region defined by the feature map calculation pattern using an integer called stride. After every calculation, the filter is moved to the right or down (when filter bounds reach right input region edge) by the stride value.

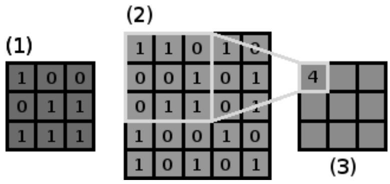


Fig. 2. Example of a feature map (3) first component calculation with a 3x3 filter (1) applied to a 5x5 data (2). The red region on the input data (2) represents the actual region, where the filter (1) is applied to produce the first component of (3)

Analyzing the presented process, the typical single channel net (i.e. CNN1) should theoretically perform better than the one we propose because of wider spatial regions of filters and so more informations might be taken into account in calculations. Despite this, our assumption is that in the case of EEG data, a convolution of a single channel portion (window) of the signal can produce more valuable information, free of noise, from other channels, in opposition to the situation where signal is treated as a whole. In our approach, the filters are set in one dimension (5x1 in the first phase and 3x1 in the second one) due to slicing process where data provided to single channel is a 16x1 vector. A single channel of data, extracted this way, represents a single frequency band channel produced from one electrode [12]. According to our assumptions, this approach does not break the core advantage of convolution process even if the single channel data region loses the multidimensional feature.

To further improve the performance of the net we performed a series of optimization tests in terms of both effectiveness (accuracy), improvement (*ACCOP* phase) and architecture optimization (*AROP* phase)

For *ACCOP* phase, in case of effectiveness improvement of learning and resulting final accuracy, during the architecture modification tests we assumed two different areas of improvement: manipulation of layers parametrization (output values, kernel size) in base function setup (PARMOD tests); testing selected learning rate modification functions for more training flexibility (LFMOD tests).

In PARMOD test series, for calibration purposes, we chose first convolution filter base size: 5x1 (CONV1), second convolution filter base size: 3x1 (CONV2) and first perceptron layer outputs number: 200 (PL1) which were compared to their starting values (Tab. 1).

From the obtained results (Table 1) we can see that decreasing the PL layer outputs did not affect the net accuracy even when the value were doubled down. As a result we adapted the doubled down value to the first PL layer which allowed to decrease the time of a single training session by nearly 15%. All time intervals

TABLE 1
Net parametrization results during PARMOD tests

Parameter	50% decrease	25% decrease	25% increase	50% increase
CONV1 filter size	-20%	-2%	-2%	-2%
CONV2 filter size	-	-	0%	0%
PL1 outputs	0%	0%	0%	-2%

for net manipulations, covered in this paper, with training time 15000 iterations, are presented in the Table 2.

TABLE 2
Global collation of experiments times

Net modification	Average training time [min]
No modification	122
Learning rate function: constant	119
Learning rate function: exponent	124
Learning rate function: sigmoid	125
Learning rate function: polynomial	125
CONV1 filter size: 3x1	119
CONV1 filter size: 4x1	121
CONV1 filter size: 7x1	123
CONV1 filter size: 8x1	124
CONV2 filter size: 4x1	121
CONV2 filter size: 5x1	122
PL1 outputs: 100	111
PL1 outputs: 150	115
PL1 outputs: 250	125
PL1 outputs: 300	128
VGG 16-layers variance	130
AlexNet variance	128
Final CNN96 architecture	98

LFMOD tests proved that step (linear) learning function was the most optimal selection in terms of learning progress. Other tested functions, which were taken into consideration, were: constant (fixed) learning, exponential change, polynomial and sigmoid. As a second phase (*AROP* phase) of the architecture enhancement we used the collected results from effectiveness improvement to assess the possible areas of optimization, maintaining the same base accuracy level with less complexity and/or less training time. Fig. 3 presents how our single channel architecture (Fig. 1 left, CNN1) changed over architecture optimization experiments, starting from the base (Fig. 3a) to the AlexNet variation [14] (3b), VGG 16-layers (further VGG-16) variation [16] (3c) and the proposed final architecture of CNN96 single channel (3d). The AlexNet and VGG-16 architectures were selected as reference, well known approaches of CNN architectures which layers arrangement represent most similar, yet different early data treating. While AlexNet uses normalization layer introducing Local Response Normalization layers to improve network generalization [14] the VGG-16 takes advantage of following convolution layers stacks with small kernel sizes to avoid overfitting [16]. Although both of those variations did not result in any improvement, the results of training time tests (Tab. 2), during which the net also presented no learning improvement, allowed to create the final approach based on double following convolution with small kernel size, 5x1 in the first convolution layer and 3x1 in the second. The last, proposed by us, approach (CNN96) resulted in less learning time and calculation complexity maintaining the same average accuracy over training sessions.

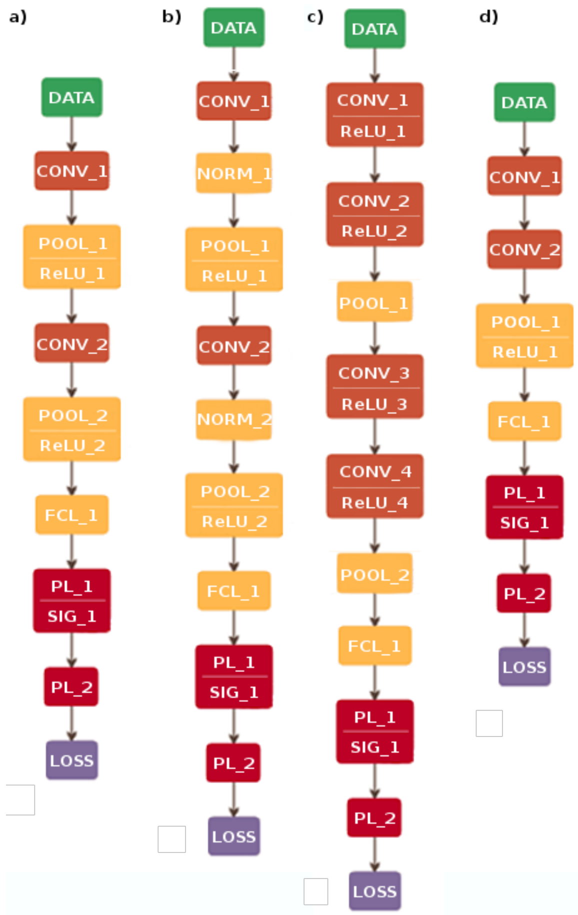


Fig. 3. The comparison of single channel approach architecture change (from the left: base CNN1 model (3a), AlexNet based variation (3b), VGG-16 variation (3c), CNN96 final approach (3d))

The main contribution of this paper, besides new architecture of CNN network, is adaptive elaboration of exploited filters characteristics. Filters are applied to the input data vector in a discrete stride value. The products of filter calculations, for a given layer, act like a matrix of features which would represent the output weights of the neurons for better activation results. In case of complex characteristics of EEG signal, and its still poorly known features correlations, we decided that weight training algorithms, that do not depend on momentum but on gradient error history, will have a critically better chance to improve the final results in terms of accuracy and training progress.

Selecting the proper solver type method for the problem, we address the general optimization problem during network update from learning accuracy by minimizing the loss factor to memory complexity. Adaptive Gradient [17] despite being also a gradient-based optimization method like SGD [18] does not rely on momentum factor. Like any other parameter history-based algorithms, instead of speeding up the training per-dimension, it computes the norm of selected previous gradients to predict the most valuable weight matrix in actual situation.

Calibration process of method hyperparameters learning was verified against modern stochastic optimization solvers: AdaGrad [17], SGD [18], AdaDelta [19] and Adam [20]. Among them AdaGrad revealed the most effective learning speed and accuracy when the net was tested against three separate subject datasets [12]. The calibration covered both time and accuracy level over certain amounts of learning time 5000, 10000 and 15000 iterations. As the Fig. 4 and 5 show, the maximum effectiveness achieved for most of experiments occurred within first 5000 iterations which in assumptions of this research is the optimal value to minimize the generalization error.

4 Results and Discussion

In order to compare the performance, we compared our methods effectiveness with other neural network based solutions results against BCI Competition III, data set V [12].

TABLE 3
Comparison of ANN based methods effectiveness

Method	Subject 1	Subject 2	Subject 3	Average
GANN	69.32	66.32	44.40	58.01
BPNN	76.02	65.89	51.14	64.34
PSONN	75.98	69.78	53.83	66.33
IPSONN	78.31	70.27	56.46	68.35
BSANN	80.32	66.03	59.34	68.56
CNN96	**81.40**	**71.95**	**55.55**	**69.63**
CNN1	78.22	62.80	52.49	64.50

For every subject the data consisted of 3 training and 1 testing file. All of the given results were calculated as an average of 5 repeats with the same NN configuration to minimize the randomization (and thus, generalization) error. The effectiveness value was calculated as a percentage of correct pattern classification against the labels provided as a 97th component in the testing file. The proposed architecture ensures an increased classification effectiveness, which proves that our method exceeds other NN methods shown in the Tab. 3.

Results presented in the Tab. 3 show that the proposed new approach involving depicted CNN architecture enhanced with adaptive gradient optimization (AdaGrad) outperforms other NN based methods used for mental tasks classification. This research show that in terms of mental task classification and overall EEG analysis for BCI usage, there is still very much to learn but results are promising for future research.

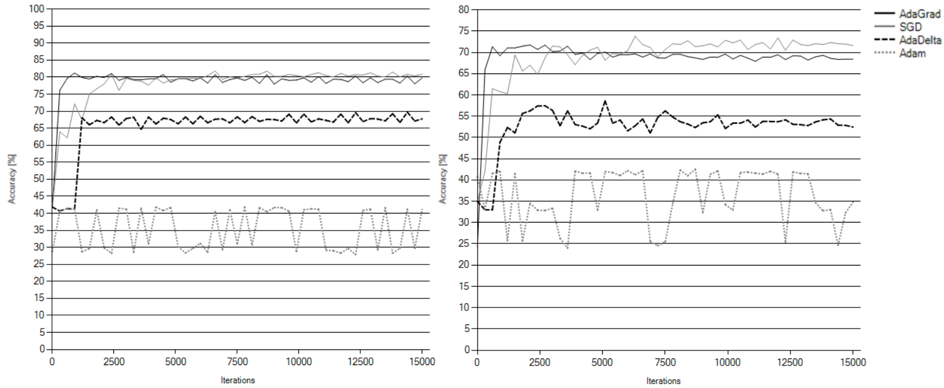


Fig. 4. Learning curves for subject 1 and 2 dataset training

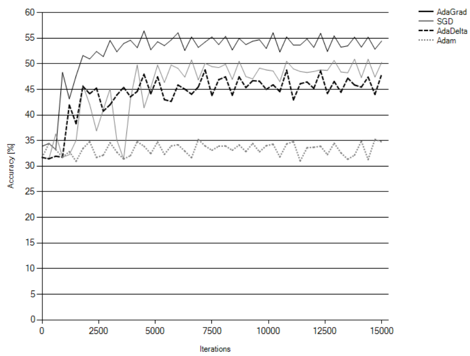


Fig. 5. Learning curves for subject 3 dataset training

References

[1] Szajerman, D.; Warycha, M.; Antonik, A.; Wojciechowski, A.: *Popular brain computer interfaces for game mechanics control*. in Multimedia and Notwork Information Systems, Advances in Intelligent Systems and Computing series, vol. 506, pp. 123-134, 2017.

[2] Sun, S.; Zhou, J.: *A Review of Adaptive Feature Extraction and Classification Methods for EEG-Based Brain-Computer Interfaces*. International Joint Conference on Neural Networks, 2014.

[3] Agarwal, S.K.; Shah, S.; Kumar, R.: *Classification of mental tasks from EEG data using backtracking search optimization based neural classifier*. Neurocomputing 166, 397-403, 2015.

[4] Lotte, F.; Congedo, M.; Lcuyer, A.; Lamarche, F.; Arnaldi, B.: *A review of classification algorithms for EEG-based brain-computer interfaces*. Journal of Neural Engineering, vol. 4, pp. 1-13, 2007.

[5] Oskoei, M.A.; Gan, J.Q.; Hu, H.: *Adaptive Schemes Applied to Online SVM for BCI Data Classification*. 31st Annual International Conference of the IEEE EMBS, USA, September 2-6, 2009.

[6] Staniucha, R.; Wojciechowski, A.: *Mouth features extraction for emotion classification*. Annals of Computer Science and Information Systems, vol.8, 1685-1692, 2016.

[7] Walczak, J.; Wojciechowski, A.: *Improved Gender Classification Using Discrete Wavelet Transform and Hybrid Support Vector Machine*. Int. Journal MGV, vol. 25, no. 1/4, pp. 27-34, 2016.

[8] Duda, R.O.; Hart, P.E.; Stork, D.G.: *Pattern Classification*. second edition, John Wiley & Sons, 2000

[9] Hasan, B.A.S.; Gan, J.Q.: *Sequential EM for Unsupervised Adaptive Gaussian Mixture Model Based Classifier*. P. Perner (Ed.): MLDM 2009, LNAI 5632, pp. 96-106, 2009.

[10] Kowalczyk, M.; Napieralski, P.; *An Effective Client-Side Object Detection Method on the Android Platform*. Journal of Applied Computer Science, Technical University Press Łódź, No. 2, pp. 29-38, 2015.
[11] Lin, C.-J.; Hsieh, M.-H.: *Classification of mental task from EEG data using neural networks based on particle swarm optimization*. Neurocomputing 72 (4-6), 1121-1130, 2009.
[12] Millán, J.R.: *On the need for on-line learning in brain-computer interfaces*. Proc. Int. Joint Conf. on Neural Networks, 2004.
[13] Galán, F.; Oliva, F.; Guàrdia, J.: *BCI Competition III. Data Set V: Algorithm Description*. 2005.
[14] Krizhevsky, A.; Sutskever, I.; Hinton, G.: *ImageNet Classification with Deep Convolutional Neural Networks*. Advances in Neural Information Processing Systems, 2012.
[15] He, K.; Zhang, X.; Ren, S.; Sun, J.: *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. Microsoft Research, 2015.
[16] Simonyan, K.; Zisserman, A.: *Very deep convolutional networks for large-scale image recognition*. ICLR, 2015.
[17] Duchi, J.; Hazan, E.; Singer, Y.: *Adaptive Subgradient Methods for Online Learning and Stochastic Optimization*. The Journal of Machine Learning Research, 2011.
[18] Bottou, L.: *Stochastic Gradient Descent Tricks*. Neural Networks: Tricks of the Trade: Springer, 2012
[19] Zeiler, M.: *ADADELTA: An adaptive learning rate method*. arXiv preprint, 2012.
[20] Kingma, D.; Ba, J.: *Adam: A Method for Stochastic Optimization*. International Conference for Learning Representations, 2015.

Dominik Szajerman was born in Łódź, Poland, in 1977. He graduated in computer graphics and obtained the Ph.D. degree from Department of Technical Physics, Information Technology and Applied Mathematics at Technical University of Lodz, Poland, in 2001 and 2010, respectively. During his pre- and postgraduate experience he was involved in a number of research projects in the areas of computer graphics, computer games and human-computer interaction. His academic interests lie in the computer graphics, GPU processing, computer games development and image processing.

Adrian Smagur Pursuing a PhD in Computer Science in Institute of Information Technology, Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology. Interested in machine learning, deep neural networks and classification with applications in areas of brain computer interfaces and computer vision.

Sławomir Opałka Game development and HCI passionate. Recieved B.S. in 2014 nad M.S. degree in Information Technology with Games and Computer Smulation Technologies specialization in 2016 from Lodz University of Technology. Fields of interest involve AI for interactive environments, machine learning methods, UX and UI design which further resulted in BCI interest. From 2016 student of PhD course at Institute of Information Technology at Lodz University of Technology with the reaserch topic of using biophysical signals for user mental state recognition.

Adam Wojciechowski PhD, DSc, Institute of Information Technology, Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology.

Supervisor of an IT teaching specialisation on Computer Games and Simulations. Since 2016 vice-Dean for Development and Business Cooperation at the Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology. Since 2016 vice-chairman of the GameInn Steering Committee in the National Centre for Research and Development. The Lodz University Rector representative in the Lodz Game Dev Cluster and Faculty Dean representative in the ICT Central Poland Cluster.

Research interests encompass: human-computer interaction, cognitive computing, computer graphics, computer vision, computer games, virtual environments, computer simulations and visualizations, computer animations, biometry and machine learning. Author or co-author of over 50 research papers. Coordinator of 3 and participant of 13 national and international research and development projects, financed by Polish and European programs.

Practical Atmospheric Scattering

Tomasz Gałaj, Adam Wojciechowski

Lodz University of Technology, Łódź, Poland
tomaszgalaj@gmail.com, adam.wojciechowski@p.lodz.pl

Abstract

We present a practical solution to implement atmospheric scattering on GPU using Compute Shaders, taking into account Rayleigh and Mie single scattering. Our implementation is based on physical equations of light transport inside the atmosphere. We show how to create simple ray-tracer on GPU and how to compute complex equation of atmospheric light scattering. This solution can dramatically decrease the time that is needed to find an approximation of integral equations used in calculating colors of the sky. We believe, this implementation can help us to seek for a new enhancements to the atmospheric scattering algorithm to be used in interactive applications such as games.

Index Terms

atmospheric light scattering, natural phenomena, participating media, GPU programming, ray-tracing, computer games, computer graphics

1 Introduction

Atmospheric scattering effects are very important for many applications that require high realism of outdoor scenes. The sky color is a natural indicator of time of day (see Figure 1).

Recent methods such as [2] or [6] calculate colors of the sky in the real time (16ms). Unfortunately, these methods make a lot of simplifications (assuming some parameters that affect the final sky color to be constant, precomputations that prevent user to change atmosphere's properties without recalculation of precomputed arrays) that tremendously affect the final result, which is far from what we can observe in real life.

In this paper, we present a practical approach to calculate atmospheric scattering, per-pixel, using OpenGL's Compute Shaders. We show how to create simple ray-tracer that is able to produce plausible sky colors in real time.

2 Light scattering

In this section we describe the physical background of atmospheric light scattering as well as mathematical model that we used in our solution.

2.1 Physical background

In computer graphics we distinguish two types of light scattering – Rayleigh and Mie scattering. Rayleigh scattering occurs for particles that are smaller than the light particles and it can be described by Condition 1.

Fig. 1. Sunset over the sea. Image courtesy of www.flickr.com

$$d \ll \frac{\lambda}{2\pi} \tag{1}$$

Where d is the particle diameter, and λ is the wavelength. On the other hand, when particles are bigger than λ Rayleigh scattering smoothly transitions to Mie scattering. This type of scattering involves aerosols as well as small steam, ice, dust particles.

The major difference between these two types of scattering is that intensity of Rayleigh scattering depends on the wavelength of scattered light, whereas Mie scattering does not. This implies that the blue sky color is the effect of Rayleigh scattering combined with a lack of violet photon receptors in people's retinas. The grey tones of halos around the sun, clouds, fog are caused by Mie scattering.

What is more, we have to define single and multiple scattering terms. Single scattering means that we take into account only one light scattering event [9] – event after which some part of light traveling towards the eye is deflected away from the viewing direction (out-scattering) or is deflected back on the viewing direction (in-scattering). On the other hand, with multiple scattering we take into account a number of light scattering events. In our solution, we take into account only the single scattering.

2.2 Mathematical model

In this section we would like to present equations of physically based mathematical model that we used in our implementation to calculate single atmospheric scattering. The equations are based on the ones that were proposed by Nishita [1].

2.2.1 Scattering intensity

Scattering intensity for Rayleigh/Mie scattering is described by Equation 2. It expresses the amount of light that has been scattered at a point P (see Figure 2) at an angle θ for a given wavelength λ.

$$I_{S_{R,M}}(\lambda, \theta, P) = I_I(\lambda)\rho_{R,M}(h)F_{R,M}(\theta)\beta^s_{R,M}(\lambda) \quad (2)$$

In the above equation, $I_I(\lambda)$ stands for the intensity of the incoming light, $\rho_{R,M}(h)$ is the Rayleigh/Mie density function, $F_{R,M}(\theta)$ is the phase function and $\beta^s_{R,M}(\lambda)$ is the scattering coefficient.

The density function $\rho_{R,M}(h)$ expresses how density of air particles decreases in dependence of altitude h. This function is defined by Equation 3.

$$\rho_{R,M}(h) = \exp(-\frac{h}{H_{R,M}}) \quad (3)$$

In the above equation, h is the altitude of the point P above the ground and $H_{R,M}$ are scale heights for Rayleigh and Mie scattering respectively. In our implementation we used $H_R = 7900m$ and $H_M = 1200m$.

2.2.2 Single scattering

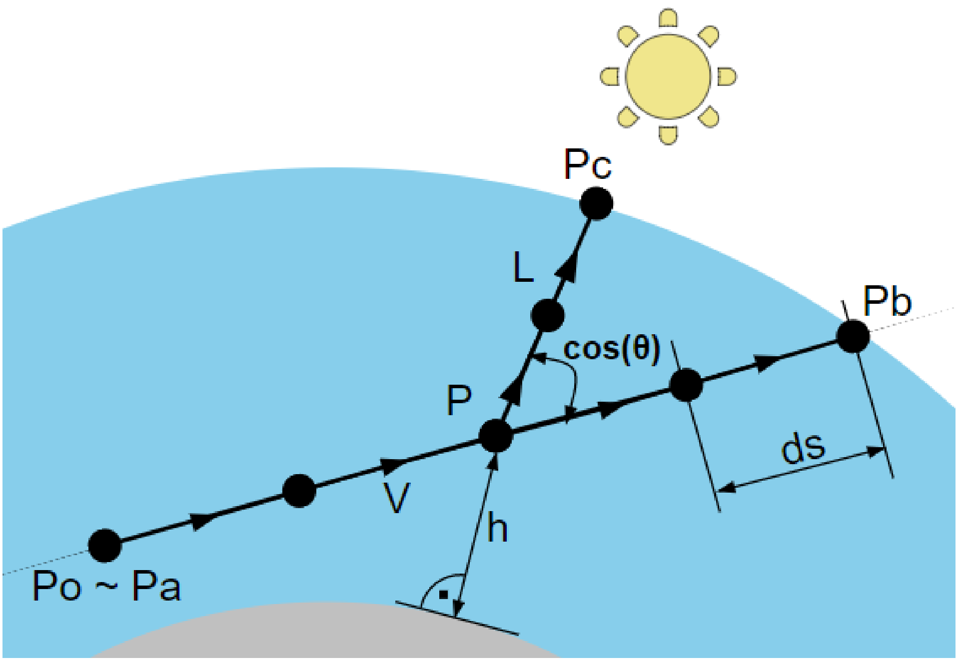


Fig. 2. Schematic of how the single scattering is being calculated

Single scattering is being computed using Equation 4 taking into account Rayleigh and Mie scattering. The result of this equation is the intensity of light that reaches the observer's eye, after exactly one scattering event.

$$I'_{S_{R,M}}(P_o, V, L, \lambda) =$$
$$I_I(\lambda)F_{R,M}(\theta)\frac{\beta_{R,M}(\lambda)}{4\pi} \cdot \int_{P_a}^{P_b} \rho_{R,M}(h)\exp(-t_{R,M}(XP_c, \lambda) - t_{R,M}(P_aX, \lambda))ds \quad (4)$$

In the Equation 4, term P_o is location of the observer's eye, V is the observer's view direction, L is the direction to the light source, λ is the wavelength of scattered light. Point P_x is the sample point at altitude h, P_a and P_b are the first and last points where density of the atmosphere is greater than zero. P_c is the intersection point of the vector coming from L and the atmosphere. We also assume that all light rays are parallel.

It should be noted, when observer is inside the atmosphere then $P_o = P_a$.

The final intensity of single scattered light I'_S is obtained by the sum of single scattering for Rayleigh and Mie. It is described by Equation 5.

$$I'_S = I'_{S_R} + I'_{S_M} \quad (5)$$

2.2.3 Optical depth

Optical depth, or transmittance expresses how much the light is being attenuated after traveling distance S in a medium. Equation 6 defines the transmittance.

$$t_{R,M}(S, \lambda) = \beta^s_{R,M}(\lambda) \int_0^S \rho_{R,M}(s')ds' \quad (6)$$

Parameter λ is the wavelength of attenuated light. Optical depth attenuation is the effect of out-scattering in participating media.

2.2.4 Phase function

The phase function is describing angular dependency of scattered light in respect to the direction of incoming light when its ray collides with a particle. This function takes as a parameter angle θ which is the angle between scattered light ray and incoming light ray. The result of this function is the amount of light that has been scattered. Equations 7 and 8 show phase functions for Rayleigh and Mie scattering respectively that were used in our solution.

$$F_R(\theta) = \frac{3}{16\pi}(1 + \cos^2(\theta)) \quad (7)$$

$$F_M(\theta) = \frac{3}{8\pi}\frac{(1 - g^2)(1 + \cos^2(\theta))}{(2 + g^2)(1 + g^2 - 2g\cos(\theta))^{\frac{3}{2}}} \quad (8)$$

Where parameter $g \in (-1; 1)$ is an asymmetry factor which describes the anisotropy of the medium (if it is forward or backward scattering). It should be noted that Mie phase function is very complex and can not be computed using single analytic function. Instead, some approximations were developed. One of the most popular approximations is Henyey-Greenstein's (Equation 8) function which is also used by our solution.

2.2.5 Scattering coefficients

Rayleigh's scattering equation provides scattering coefficients for a volume for which we know its molecular density. These coefficients are given by Equation 9.

$$\beta_R^s(\lambda) = \frac{8\pi^3(n^2-1)^2}{3N\lambda^4} \tag{9}$$

Where superscript S indicates that it is a scattering coefficient, subscript R means that it is a coefficient for Rayleigh scattering, λ is the given light's wavelength, n is the index of refraction of air and N is the molecular density at sea level. As we can see, this equation depends on the wavelength. If the wavelength is short, the value of β_R^s will be higher and if the wavelength is long, the value of β_R^s will be lower.

This explains why the sky is blue during the day and red-orange at the dawn or dusk. During the day, when the sun is in the zenith, light has relatively short distance to travel before reaching observer's eye – blue light is being scattered towards the eye and sky appears blue (green and red light need to travel bigger distance to by scattered). On the other hand, at sunrise or sunset, light has to travel much longer distance than in the previous scenario. Before light reaches the observer's eye, most of the blue light has been scattered away, and only red-green light will have a chance to reach observers eye.

Mie scattering is similar to Rayleigh, but it applies to particles that are much greater than the scattered wavelength. These particles (aerosols) may be found on low altitudes of the Earth's atmosphere. Therefore, equation for Mie scattering coefficients needs to be slightly different (Equation 10).

$$\beta_M^s() = \frac{8\pi^3(n^2-1)^2}{3N} \tag{10}$$

Where subscript M stands for Mie scattering the rest of the parameters are the same as in the Rayleigh scattering coefficient equation.

3 RELATED WORK

Simulation of realistically looking atmosphere was being studied by computer graphics researchers for a long time. Due to the fact that, physically based equations, that are required to compute atmospheric scattering are tremendously time-consuming (≈10s per frame on i5-4690, GTX 970), it was not possible to perform this operations in real time. Therefore, first simulations were performed off-line. In the following subsections we present related work to the subject that we split in four categories: the earliest models, analytic models and GPU based models.

One of the first papers tackling the subject of simulating atmospheric scattering was [1]. Nishita et al. presented a method for calculating atmospheric scattering taking into account Rayleigh and Mie single scattering. Their method was based on a set of physically based equations for calculating single scattering in the atmosphere. This solution is being used until today, however it does not enable user to change atmosphere's density (e.g. to visualize different planet's sky) due to pre-computed values in a lookup table (that speeds up computations). Nishita et al. also presented an equation for calculating ambient illuminance of the ground and oceans. This volumetric based implementation was capable of rendering Earth's atmosphere only from space (Figure 3).

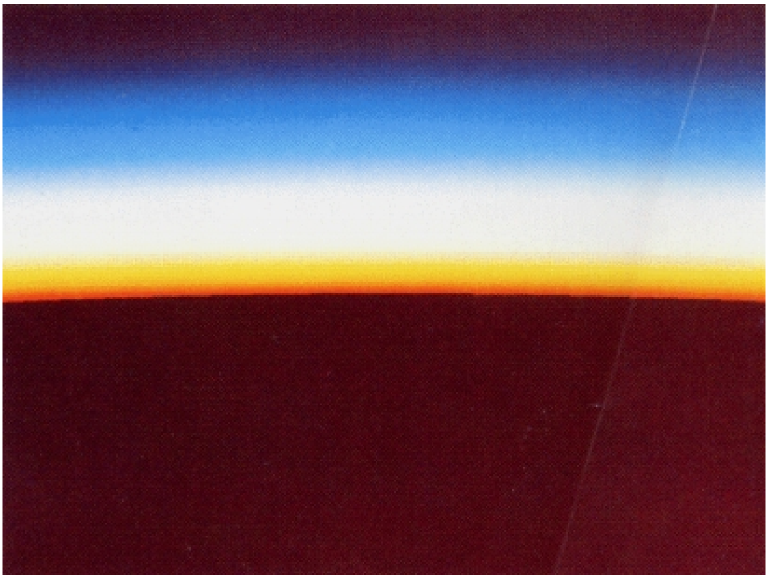


Fig. 3. Nishita's atmospheric scattering simulation [1]

One of the most widely used analytic model is Preetham's model [6] introduced in 1999. His solution was able to calculate colors of the sky in the real time, but had many major drawbacks (it breaks down numerically, unrealistic luminance distribution) as was pointed in [7]. In some specific conditions Preetham's model was giving negative values of intensity and under some circumstances the model was behaving incorrectly [4], [7].

In 2005, O'Neil [2] presented one of the first implementations of atmospheric scattering on GPU. His solution was able to perform integration in real time (Figure 4) per vertex instead of per-pixel; he was doing all of the calculations in Vertex Shader. However, it was a rough approximation since he was using analytic approximation of transmittance (optical depth) function instead of pre-computed lookup table (texture fetches from Vertex Shader were not allowed). This function worked only for one specific scale height (a distance over which an atmosphere's density decreases by a factor of e) value. Therefore, if a user would like to use a different value for scale height, s/he has to perform complex calculations to obtain the new analytic formula for the new scale height value. He also proposed an idea that single scattering values might be stored in a big 3D lookup table, but he was not able to make his idea real.

In 2012, Wojciechowski [8] used another approach to simulate sky colors in real-time. His solution was designed for games and worked very well. However, this implementation also used a lot of simplifications that affected the realism of the final image.

The first method that used pre-computed single scattering values in real time was Schafhitzel et al. [5]. The authors succeeded to precompute single scattering values in a big 3D lookup table as O'Neil suggested [2]. Later, it turned out that this 3D texture was still lacking one dimension.

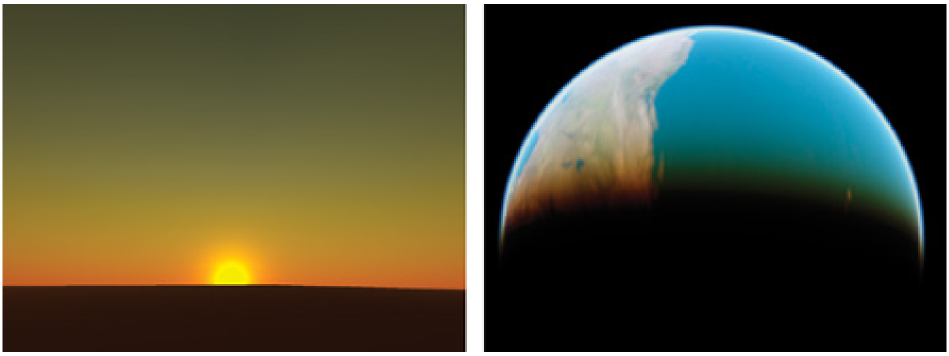


Fig. 4. O'Neil's atmospheric scattering simulation [2]

In 2008, Bruneton et al. in paper [3] presented a method that based on Schafhitzel's work. They presented the first real time method that was taking into account multiple scattering and they used pre-computed 4D lookup table. Their method was working for all viewing directions and virtual camera positions at the daytime.

In the following sections we will describe our solution that utilizes GPU to compute atmospheric scattering with accuracy on a per-pixel level.

4 Solution description

Our implementation is based on physical equations (Section 2.2) of atmospheric light scattering that the most accurately describes the reality e.g. Figure 1. We are not creating any lookup tables with pre-computed values. Instead, we are numerically calculating the value of the integral showed in Equation 4 and Equation 5. Numerical method that we use is Riemann Sum.

In the following subsection we first describe the host application that prepares all the necessary data for Compute Shader on CPU (host) side and then passes these data to GPU. Next, in Compute Shader section we describe the whole process of generating colors of the sky using GPU based backward tracing algorithm (as depicted in Figure 5). Before mentioned pseudo code is showed in Listing 1.

4.1 Host application

To be able to perform backward tracing (Figure 5) algorithm on GPU we have to prepare some resources on the CPU side. First, we create Texture Object (TO) that will be used by Compute Shader (CS). This TO will be referred as an output image to which CS will be writing computed pixels. Moreover, this TO should have at least the same dimensions as the viewport to avoid pixelization of the ray-traced image.

Then CS is being responsible to perform backward tracing algorithm in order to produce colors of the sky from the given perspective. The output of the CS is being saved to TO that was created earlier on the CPU side. Finally, we draw a full screen quad textured with TO that holds the pixels that CS has generated. Above mentioned process is showed in Listing 2.

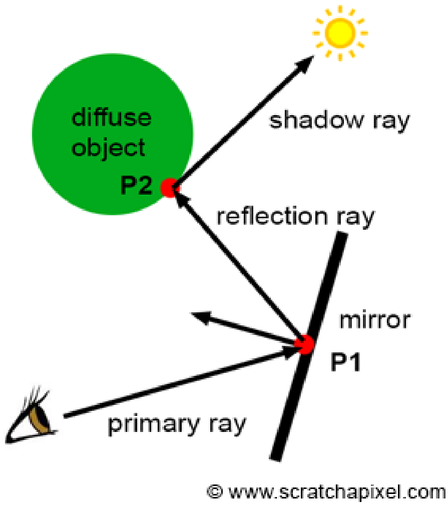


Fig. 5. Scheme of backward tracing algorithm. Image courtesy of www.scratchapixel.com

Listing 1 Backward tracing algorithm

```
for every pixel do
  Cast a ray from the eye – primary ray
  for every object in the scene do
    Find intersection with the primary ray
    If found intersection is the closest, keep it
  end for
  for every light in the scene do
    At intersection point cast shadow ray to the light source
    if shadow ray did not hit anything then
      Compute color at the closest found intersection point
      If object has reflective surface cast reflection ray and compute color for
      nearest intersection point
    end if
  end for
end for
```

4.2 Compute Shader

In order to compute atmospheric scattering using backward ray-tracing algorithm we need to create Compute Shader. We will only show the function that is responsible for calculating atmospheric scattering. Its pseudo-code is shown in Listing 3. The rest of the remaining functions that build Compute Shader are just helper functions required by any ray-tracer i.e. function for calculating *primary ray* (Figure 5).

First of all, we define inputs for our single scattering function which are *primary ray* that comes from standard ray-tracing computation of the viewing ray from the virtual camera to the specific pixel. The next two inputs d_{min} and d_{max} are distances

Listing 2 Host application preparations

1: Create *output image* TO
2: Dispatch Compute Shader with backward tracing algorithm to compute atmospheric scattering
3: Bind *output image* TO as a texture 2D
4: Draw contents of *output image* onto full screen quad

Listing 3 Compute Shader Atmospheric Scattering

Require: $primary\ ray, d_{min}, d_{max}$
Ensure: $d_{min} < d_{max}$
1: $num\ samples = 16$
2: $num\ light\ samples = 8$
3: $segment\ length \leftarrow \frac{d_{max} - d_{min}}{num\ samples}$
4: Calculate $phase_R$ using Equation 7
5: Calculate $phase_M$ using Equation 8
6: **for** $i = 0$ to $i < num\ samples$ **do**
7: Calculate sample position using primary ray data
8: Calculate sample height using sample position and planet's center
9: Compute part of optical depth H_R and H_M using Equation 3
10: $optical\ depth_R \leftarrow optical\ depth_R + H_R$
11: $optical\ depth_M \leftarrow optical\ depth_M + H_M$
12: Compute ray to light using sample position and sun position
13: **for** $j = 0$ to $j < num\ light\ samples$ **do**
14: Calculate sample light position using light ray data
15: Calculate sample light height using sample light position and planet's center
16: Compute part of light optical depth HL_R and HL_M using Equation 3
17: $optical\ light\ depth_R \leftarrow optical\ light\ depth_R + HL_R$
18: $optical\ light\ depth_M \leftarrow optical\ light\ depth_M + HL_M$
19: **end for**
20: Calculate transmittance using Equation 6
21: **end for**
22: Calculate final single scattering intensity using Equation 4

from the camera along the primary ray, where were detected intersection points with the planet and/or its atmosphere.

Next, in lines 4 and 5 we calculate the value of the phase functions for Rayleigh and Mie scattering.

In line 6 we define the for loop that will sample (step-by-step) the viewing ray, computing the out-scattering factor – transmittance. In this loop, we define another for loop (line 13) that is responsible for calculating in-scattering factor – how much sun light is added to the specific sample.

Finally, in line 20, we calculate optical depth (transmittance) and in line 22 we use Equation 4 to get the final value of single scattering across all viewing samples.

5 Results

The following results were obtained using NVIDIA GeForce GTX 950M GPU and Intel Core i5 4200H CPU and the rendered images' resolution was fixed to 640x480. Figure 6 shows the result that was rendered using our implementation of simple ray-tracer on GPU using Compute Shader.

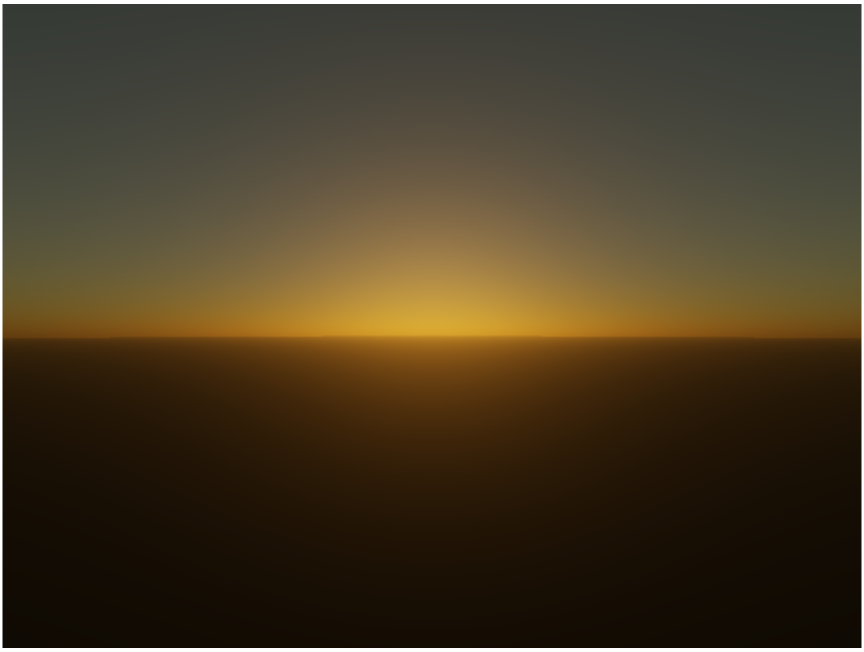


Fig. 6. Sunset generated with GPU based ray-tracer

On the other hand, Figure 7 shows performance differences between CPU and GPU version of the algorithm. Using GPU technique we were able to render one frame in 14.765 ms, where rendering the same image, using the same multi-threaded algorithm on CPU took about 3s.

6 Conclusions

We observed that harnessing power of the GPU can speed up our research despite the fact that it takes slightly more time to transfer the algorithm from the CPU to the GPU version. We no longer have to waste time to wait for one image to render for 3 s, because now we can generate over 60 images per second. This way we can seek for a new enhancements to the atmospheric scattering to reduce rendering time below 1.5 ms. What is more, our solution does not use any simplifications to the physical model and we do not use any pre-computed lookup tables to speed up the calculations. Moreover, the accuracy of our solution is per-pixel.

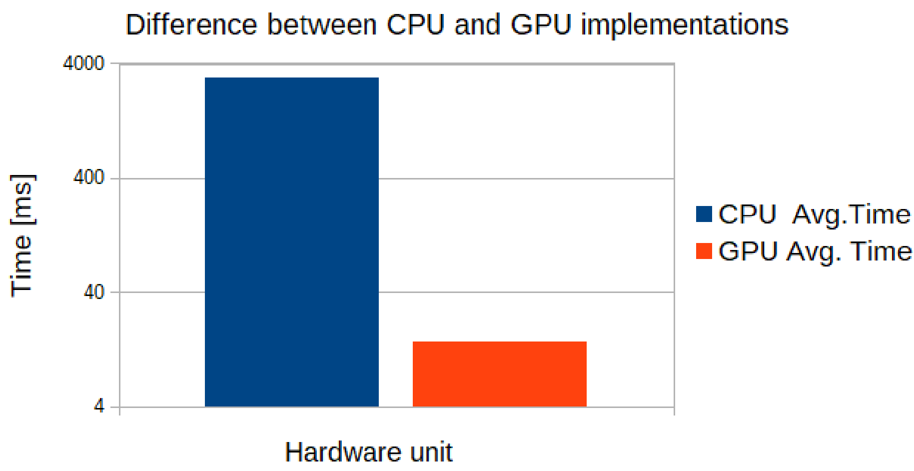


Fig. 7. Comparison between CPU and GPU ray-tracing atmospheric scattering

References

[1] Sirai, T.N.T.; Nakamae, K.T.E.: *Display of The Earth Taking into Account Atmospheric Scattering*. In *SIGGRAPH 93: Conference Proceedings, August 1-6, 1993* (p. 175). Addison-Wesley Longman, 1993.
[2] O'Neil, S.: *Accurate atmospheric scattering*. GPU Gems, 2, pp. 253-268, 2005.
[3] Bruneton, E.; Neyret, F.: *Precomputed atmospheric scattering*. In Computer Graphics Forum, vol. 27, No. 4, pp. 1079-1086, Blackwell Publishing Ltd., 2008.
[4] Bruneton, E.: *A Qualitative and Quantitative Evaluation of 8 Clear Sky Models*. IEEE transactions on visualization and computer graphics, 23(12), pp. 2641-2655, 2017.
[5] Schafhitzel, T.; Falk, M.; Ertl, T.: *Real-Time Rendering of Planets with Atmospheres*. In Journal of WSCG, volume 15, 2007.
[6] Preetham, A.J.; Shirley, P.; Smits, B.: *A Practical Analytic Model for Daylight*. In Siggraph '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pp. 91-100, 1999.
[7] Zotti, G.; Wilkie, A.; Purgathofer, W.: *A Critical Review of the Preetham Skylight Model*. In WSCG 2007 Short Communications Proceedings I, pp. 23-30, 2007.
[8] Wojciechowski, A.: *Real time sky visualization for computer games*. In Computer Science in Novel Applications, Editors: Romanowski A., Sankowski D., subtitle: "Human-Centered Interdisciplinary Applications", publication series: Lodz University of Technology Monograph, Łódź, pp.125-136, 2012.
[9] Jarosz, W.: *Efficient Monte Carlo Methods for Light Transport in Scattering Media*. UC San Diego PhD thesis, September 2008.

Tomasz Gałaj PhD student at Institute of Information Technology, Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology.

Author of the Real-time Rendering class blog and Vertex Engine for proof of concept applications. Research interests comprise: computer graphics and animation, computer games, computer simulations. Participant of 2 national research and development projects, financed by Polish and European programs.

Adam Wojciechowski PhD, DSc, Institute of Information Technology, Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology.

Supervisor of an IT teaching specialisation on Computer Games and Simulations. Since 2016 vice-Dean for Development and Business Cooperation at the Faculty of Technical Physics, Information Technology and Applied Mathematics, Lodz University of Technology. Since 2016 vice-chairman of the GameInn Steering Committee in the National Centre for Research and Development. The Lodz University Rector representative in the Lodz Game Dev Cluster and Faculty Dean representative in the ICT Central Poland Cluster.

Research interests encompass: human-computer interaction, cognitive computing, computer graphics, computer vision, computer games, virtual environments, computer simulations and visualizations, computer animations, biometry and machine learning. Author or co-author of over 50 research papers. Coordinator of 3 and participant of 13 national and international research and development projects, financed by Polish and European programs.

Face navigation assistant supporting people with visual impairment

Michał Kowalczyk, Piotr Napieralski

Lodz University of Technology, Łódź, Poland
piotr.napieralski@p.lodz.pl

Abstract

This paper focuses on mobile systems that facilitate the social functioning of disabled people. It particularly describes innovative assistive applications intended for people with visual impairment. The study contains a brief description of face navigation process and tools allowing implementation of this issue in the Android environment. The result of this paper is a self-developed application which enables recognizing people by using the Android device. The application provides the ability to lead the way to the selected person based on the camera view. It can be used by people with visual impairment in the case of doubt about the identity or location of the person in their immediate vicinity.

Index Terms

mobile systems, visual impairment support, assistive technology, Android application

1 Introduction

Modern technologies provide a wide variety of tools enabling human-computer interaction. This special communication is getting more and more advanced [1]. It means development in the systems and techniques of information exchange, so that the interactions are becoming surprisingly natural and intuitive [2]. The interfaces utilised in the communication process between IT system and a customer implement the mechanism of the synergy between digital information and human senses. The classical application interfaces engage primarily three human senses: the touch, which provides the ability to input the data to the program, as well as the sight and the hearing enabling acquisition of output information from the IT system [3]. The set of instructions interchanged between an user and a device with the aid of the above mentioned mechanisms provides symmetric communication, which enables people to control the typical computer application [4].

The described model presents the standard situation, in which the customer does not suffer from any handicaps and health limitations. It has to be noticed that nowadays people with visual or hearing impairment are accounting for the still increasing percentage of the users of different kinds of applications and computer programs. The trend is conditioned by both the high availability of ICT devices and the expanding problem of diseases of civilisation. Currently it is evaluated that over 300 million people worldwide are afflicted by hearing or visual impairment [5] [6]. About 54% of the blind and 69% of the deaf people use smartphones [7]. Moreover the researches indicate that above mentioned diseases have been becoming more and more commonplace for recent years. To make things worse the scientists speculate that the trend will sustain [8] [9]. The enumerated factors suggest the increasing demand for the digital solutions, which are adapted or dedicated for people with health limitations. The above mentioned issue concerns the development of customer software.

In the process of developing applications for disabled people it is significant to provide a common runtime environment and simplicity of the software distribution. The mobile systems fulfil the above mentioned criteria unbeatably. There are over 3 billion actively utilised devices (at least once a month), which have installed mobile systems [10]. Android is the most popular mobile platform worldwide. The number of active devices with this environment is estimated at 2 billion which represents about 70% of all mobile operating systems available on the whole market. The distribution of the software on the Android platform is realized in the standardised way with the aid of Google Play channel, which is an easy and safe mechanism of delivering an application to the customers. As a result the Android platform is marked by high availability and convenience of exploitation, which makes it highly attractive development environment.

2 Visual Impairment Support

Programs which fulfil the function of assistants constitute a prominent group of applications dedicated for mobile devices [11] [12]. The assistants are information technology solutions which support social functioning of disabled people [13] [14]. The tasks performed by the discussed technology have a wide variety of applications, ranging from simple auditory issues to highly advanced analytic solutions. The assistant tools enabling image processing provide a significant set of functionalities dedicated for people with visual impairments. Three basic categories may be distinguished among the applications for the visually impaired [15]. The first one involves solutions from the scope of text and audio analysis forming the transcription systems. The second set consists of the applications providing navigation functions for people with visual impairment [16]. The last group includes programs processing the camera view and voice informing the user about the features of the surroundings [17] [18]. Generated sound messages define objects and people detected in the analised view. The obtained information enable enhancing security of the disabled user.

Among many analysed applications it was not successful to find a software enabling simultaneous identification and leading the way to the person situated in the camera view of the mobile device. The stated solution would enable the visually impaired to move autonomously and to avoid problems resulting from incorrect identifying a person. The mechanism would not only increase the convenience of everyday life of disabled people, but most of all it would largely enhance their security. Due to the above it was decided to realize mentioned idea by implementing an appropriate mobile application.

3 Face Navigation

Functionality of identifying and navigating to the chosen person may be defined by the term face navigation [19]. The mentioned mechanism implements three main algorithm issues: face recognition, navigation and speech synthesis (Fig. 1).

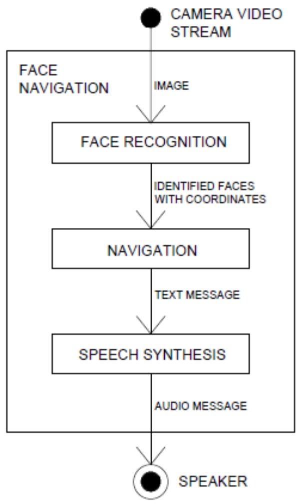


Fig. 1. Face navigation block diagram

The process of face navigation works sequentially starting from retrieving an image frame from the video stream. The extracted view shall be subject to the segmentation of the image in the context of the face recognition mechanism. The received data relevant to identified people and their location are passed to the navigation component. All the provided digital information is converted to the text guidelines, which enable the user to figure out the proper direction towards the chosen person. The next step consists of the speech synthesis of the generated messages, which enables audio reproducing of the information obtained in the process of face navigation.

3.1 Face Recognition

The face recognition is an issue from the scope of image processing which enables establishing the identity of the people situated in the camera view. The process of the identification consists of the biometric analysis of the image within the context of faces' database [20]. In the framework of the recognition mechanism the consecutive computational stages can be marked out (Fig. 2).

The input image initially shall be subject to the detection process. The aim of this step is to detect and determine the coordinates of the faces in the analysed view. Commanding the input image frame and the information about faces locations it is possible to separate subimages representing the effigies of people detected in the photo. The received images are subject to extraction enabling the unambiguous description of the analysed photograph in the numerical space. The result of the

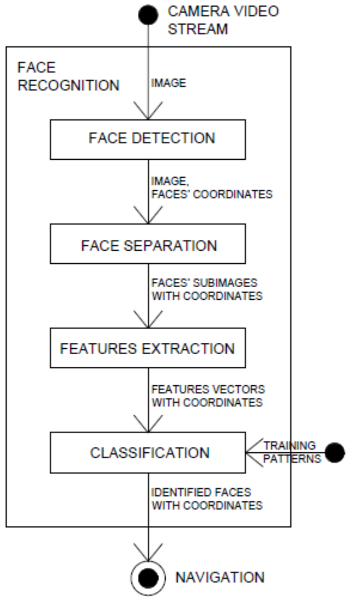


Fig. 2. Face recognition block diagram

process of marking out features is usually a binary vector containing the information about the characteristics of changes in brightness of the processed image. It has to be highlighted that the extraction method utilised in the process should be coherent with comparative vectors in the database. The assumption enables carrying out the clasification process of matching the training patterns with the features′ vectors extracted from the analysed photograph. As a result of the above described phase it is possible to obtain the identity of the person recognised in the image and the coordinates enabling locating the face in the analysed space.

3.2 Navigation

Marking out the instructions enabling the user to move towards the chosen person is based on analysis of the position of the face in the picture. The localisation of the recognised effigy in the image in relation to the device determines referential direction of movement enabling the user to get to the selected person. The navigation process divides the image into separate rectangular areas which are closely related to a specific direction instruction. Accuracy of the calculated direction depends on the amount of zones considered in the full image. The large amount of the areas provides more precise directions which on the other hand requires more complicated model of the view processing mechanism. In the case of typical uses the number of zones stands at about 12 assuming the Cartesian distribution 4 (horizontally) $\times$ 3 (vertically). Assigning the person to the specific area is conditioned by the absolute coordinates of the subimage center representing the position of the face in the camera view. The face center is calculated on Cartesian system as the arithmetical mean of absolut boundary coordinates of the subimage isolated in the process of face separation. The navigation direction is determined by intersection of the face center and the specific zone's boundaries. The generated navigational instruction indicates a suggestion to move in a proper direction. The number of possible messages is equal to the amount of implemented zones which directly translates into the navigational information in the form of guidelines for example such as 'slightly left', 'straight ahead' or 'sharp right' depending on the classifying the location of the person to the specific zone of the image. The navigational instructions are sequentially estimated for each image processed in face navigation mechanism. The generated messages are in the text form.

3.3 Speech Synthesis

The text navigational instructions require converting character notation into an audio form. The mentioned functionality is realized by the process of speech synthesis. The mechanism enables running the transformation of text message into the human speech. The process of synthesis starts from the meaning analysis of the material, which enables separating linguistic information. The next step is running the transcription converting the linguistic content into phonetic form. Phonemes with the prosody are subject to decoding and concatenation process with the aid of elementary sounds enabling the synthesis of a sound wave. The generated sound wave adjusted in a correction phase allows getting a digital voice message characterised with a proper intonation and fluency. The obtained track played with the aid of a loudspeaker represents a text content in the form of synthesised speech which is fully understandable for a human being.

The instructions generated in the process of speech synthesis present data about people identified in the picture and their localization. The audio form of presenting communications enables the users with visual impairments to receive full information concerning the data obtained in the process of face navigation.

4 APPLICATION

The implementation of the presented model (Fig. 1) forms a realization of the mobile assistant dedicated for the visually impaired called Face Navigator. The application has been implemented in the JAVA environment utilising the set of programistic tools available in the framework of the Android SDK package (Software Development Kit). Android SDK provides a range of generic solutions simplifying programmers to design, test and implement mobile applications.

4.1 Implementation

The architecture of the application Face Navigator implements basic MVC structure (Model-View-Controller) and utilises mechanisms available in the set of Android SDK tools. The view layer is based on a generic component (*android.hardware.Camera*) processing the real time camera view and the messages obtained in the face navigation process. Data model used in the application has a standardised schema adjusted to the raised issues utilising mechanisms of the JAVA programming environment. The elements of data composing a training set of faces are loaded in the application through the Facebook social network utilising the external library Facebook API (Application Programming Interface). The application controller processing data from a camera stream is based on the activity mechanism (*android.app.Activity*) available in the Android SDK framework. The process of face navigation ran in the main activity of the application uses primarily the algorithms derived from Android SDK and the external image processing library called OpenCV.

The implementation of the face navigation mechanism in the Android environment starts from the phase of initial image preparation. The frame obtained from the camera stream in the YUV colour model requires converting into RGB form. Process of the transformation YUV – RGB consists of multiplying every YUV pixel of the image by the the matrix of coefficients describing correlation between YUV and RGB spaces (eq. 1).

$$\begin{bmatrix} 1 & 0 & 1.13983 \\ 1 & -0.39465 & -0.58060 \\ 1 & 2.03211 & 0 \end{bmatrix} \cdot \begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

The image representing a camera view in the RGB model is utilised in the process of face detection. The Android SDK tools provide ready and complete method and a set of data structures (*android.media.FaceDetector*) enabling determining position of the face in the picture. The function *findFaces* receiving an image frame returns the collection of absolute coordinates corresponding to detected effigies. The process of face identification, which is much more complicated issue, is not natively supported by mechanisms of Android environment. Therefore, it is essential to use an external package from the field of image processing which is compatible with the Android platform. The OpenCV library including a set of complete face identification tools complies with the above mentioned requirements. The Eigenfaces method is relatively a robust algorithm of face recognition implemented in OpenCV framework. It provides the ability to extract the identification features representing the face in the analysed picture and to compare it with the data stored in the database [21]. The similarity verification process of the features collection's pairs was made by the Brute-Force method utilising Euclidean metric [22]. The minimal value of the calculated metric which satisfies the boundary condition of the classification means the identification of the person in the analysed image space. The information obtained in the process of recognition and the parameters of corresponding faces are passed to the navigation process. The navigational algorithm is based on self-developed solution according to the presented way (chap. 3.2). Afterwards the generated text messages are processed by the mechanism of speech synthesis with the aid of the component belonging to Android SDK (*android.speech.tts.TextToSpeech*). The utilised text-to-speech tool enables convert the character string into audio form that can be presented by the loudspeakers on the mobile device.

The Face Navigator application implemented in the Android environment according to the above presented algorithm is a ready solution, available to use by people with visual impairment. The program does not require any advanced configurations or additional mechanisms apart from the typical hardware interfaces available on devices dedicated for mobile platforms. After installing the Face Navigator automatically complements the necessary third-part software like OpenCV framework. Starting the application runs the cyclic process of face navigation. The Face Navigator output informations are presented sequentially on the screen and by the loudspeakers informing the user about people being in the immediate vicinity (Fig. 3).

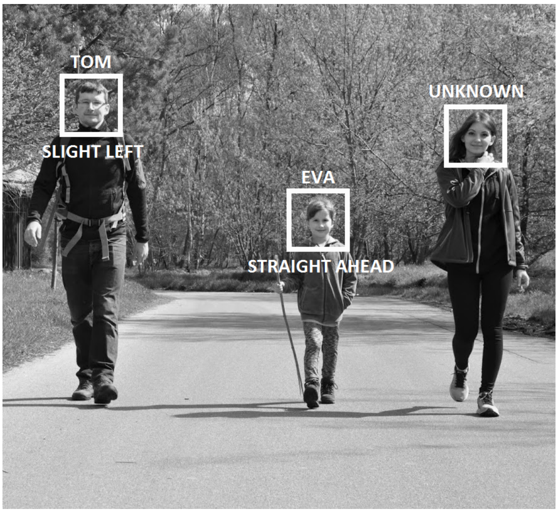


Fig. 3. The Face Navigator application view – analysing from the left: person 1 – TOM: SLIGHT LEFT; person 2 – EVA: STRAIGHT AHEAD; person 3 – UNKNOWN

4.2 Tests

The application evaluation requires running a set of tests determining correctness of the generated results depending on the surrounding's parameters. The distance between the camera and the processed object, orientation of the face in the view and the number of patterns used to build a training database are the major factors which have an influence on effectiveness of the developed application.

The initial test phase consists of determining the referential level of correct identifications generated by the application. The correct identification may be defined as the full accordance between the person identity specified in the face navigation process and the database image label. Analysis of the maximum effectiveness of the developed process requires laboratory test conditions. The results generated in favourable environment form a comparative criterion for the application evaluation. The test process was developed as a set of 100 consecutive attempts during which it was verified the correctness of identity identification of 20 people presented on the images (2 of them not included in the testing database). The database utilised in the research was made of eighteen effigies of people from the database of faces AT&T Cambridge (the database contains ten different images of each person) [23]. The research process was made by using two devices Samsung Galaxy S3 (8 Megapixel camera) and Samsung Galaxy Tab2 (3.2 Megapixel camera) both with the preinstalled operating system Android v.4.0. The driven tests made it possible to conclude that the Face Navigator application can achieve even 91% operations' efficiency in favourable environmental conditions.

The information about the highest application accuracy makes it possible to carry out comparative tests in real conditions. The carried out studies indicates a strong dependence of correctness of the application's results and the runtime environment. The factors, for which the tests were conducted, were analysed separately.

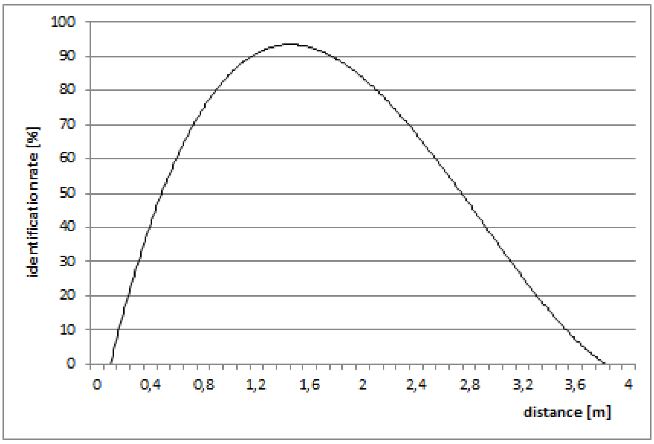


Fig. 4. The mean percentage rate of correct identifications of people in relation to the camera distance

In the case of the distance influence on generated results correctness polynomial relationship of this factor and the number of accurate identifications may be noticed. The best results (over 70%) were achieved within the distance range of 1 to 2.5 meters (Fig. 4). Beyond the mentioned values the decline of correctness of the applications occurs.

The face orientation in the analysed image is also an essential factor. The characteristic of the results welcomes the normal distribution trend with the highest efficiency (over 70%) within the range of 0-25 degrees of deviation from the norm of direct view (Fig. 5). The straight-ahead face orientation constitutes the most prominent case for achiving positive identifications' rate.

The amount of training patterns is a little bit less crucial, but still determines strongly the correctness of generated identifications. Large number of elements of the training database improves application operations. The minimum number of patterns per person which enables providing correct results (over 70%) shall be 3, whereas the best results are achieved when the application uses 10 training elements (Fig. 6).

The conducted test process indicates that the developed algorithm realize correctly the face navigation function. The above analysis of the influence of the external factors on the efficiency of the application presents the results typical for the issue of people detection and identification in the picture. The achieved results confirm the applicability of the described mechanism as an commercial assistant for people with visual disparities.

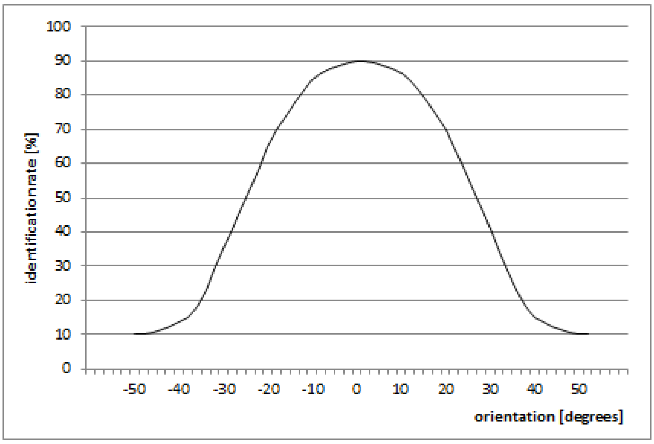


Fig. 5. The mean percentage rate of correct identifications of people in relation to the face orientation in the picture

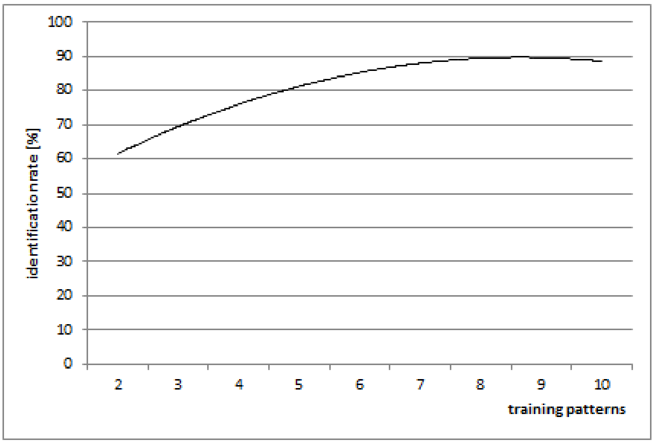


Fig. 6. The mean percentage rate of correct identifications of people in relation to the amount of training patterns

5 Conclusions

The realized considerations proclaim the high demand for applications dedicated for disabled people. The visually impaired are a large group of consumers of the assistant software. The applications fulfilling the specialised functions may be realized by the aid of standard mobile platforms. Popularity of the runtime environment enables providing the software to the largest amount of customers. Among mobile platforms the market was predominantly captured by the Android system providing an intuitive and well-equipped development environment. The process of developing programmes dedicated for the Android platform is much easier due to the multitude of tools available for free as a part of the SDK package.

The implemented application Face Navigator, realizing the described process, is a stable solution enabling the visually impaired to identify people in their immediate vicinity. The system provides also the function of audio navigating towards the recognized people. It is evaluated in the testing process that the operations of the developed mechanism achieve satisfying results providing 91% correct identifications in the optimistic case and up to over 70% assuming negative parameters of the surroundings. The face navigation is an innovative idea, which has no found equivalents available on the mobile application market.

REFERENCES

[1] Lazar, J.; Feng, J.; Hochheiser, H.: *Research Methods in Human-Computer Interaction*. Second Edition, Morgan Kaufmann, 2017.

[2] Szajerman, D.; Warycha, M.; Antonik, A.; Wojciechowski, A.: *Popular Brain Computer Interfaces for Game Mechanics Control*. Multimedia and Network Information Systems: Proceedings of the 10th International Conference MISSI 2016, pp. 123-134, 2017

[3] Densmore, M.; Smyth, T.N.; Kam, M.; Dearden, A.: *Human-Computer Interaction for Development: The Past, Present, and Future*. Information Technologies and International Development vol. 5, pp. 1-18, 2009.

[4] Barbosa, S.D.J.; Lazar, J.: *Introduction to Human-Computer Interaction*. Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, pp. 1202-1204, 2017.

[5] Pascolini, D.; Mariotti, S.P.: *Global Estimates of Visual Impairment: 2010*. British Journal of Ophthalmology vol. 96, pp. 614-618, 2012.

[6] Olusanya, B.O.; Neumann, K.J.; Saunders, J.E.: *The Global Burden of Disabling Hearing Impairment: A Call to Action*. Bulletin of the World Health Organization 2014 vol. 92, pp. 367-373, 2014

[7] Morris, J.; Mueller, J.: Blind and Deaf Consumer Preferences for Android and iOS Amartphones, Inclusive Designing, pp. 69-79, 2014.

[8] Stevens, G.A.; White, R.A.; Flaxman, S.R.; Price, H.; Jonas, J.B.; Keeffe, J.; Leasher, J.; Naidoo, K.; Pesudovs, K.; Resnikoff, S.; Taylor, H.; Bourne, R.R.A.: *Global Prevalence of Vision Impairment and Blindness: Magnitude and Temporal Trends, 1990-2010*. Ophthalmology vol. 120, pp. 2377-2384, 2013.

[9] Varma, R.; Vajaranant, T.S.; Burkemper, B.; Wu, S.; Torres, M.; Hsu, C.; McKean-Cowdin, R.: *Visual Impairment and Blindness in Adults in the United States: Demographic and Geographic Variations from 2015 to 2050*. JAMA Ophthalmology vol. 134, pp. 802-809, 2016.

[10] Maxl, E.; Döring, N.; Wallisch A.: *Mobile Market Research: Analysis through the Mobile Phone*. Herbert von Halem Verlag, 2009.

[11] Hakobyan, L.; Bartlett, H.; Lumsden J.; O'Sullivan, D.: *Mobile Assistive Technologies for the Visually Impaired*. Survey of Ophthalmology vol. 58, pp. 513-528, 2013.

[12] Liu, X.; Doermann, D.; Huiping, L.: *Mobile Visual Aid Tools for Users with Visual Impairments*. Mobile Multimedia Processing, Springer-Verlag Berlin, pp.21-36, 2010.

[13] Pal, J.; Pradhan, M.; Shah, M.; Babu R.: *Assistive Technology for Vision-impairments: An Agenda for the ICTD Community*. Proceedings of the 20th International Conference Companion on World Wide Web 2011, pp. 513-522, 2011.

[14] Capi G.; Toda H.: *A New Robotic System to Assist Visually Impaired People*. In the 20th IEEE International Symposium on Robot and Human Interactive Communication 2011, pp. 259-63, 2011.

[15] Csapó, A.; Wersényi, G.; Nagy, H.; Stockman, T.: *A Survey of Assistive Technologies and Applications for Blind Users on Mobile Platforms: A Review and Foundation for Research*. Journal on Multimodal User Interfaces vol. 9, pp. 275-286, 2015.

[16] Baranski, P.; Polanczyk, M.; Strumillo P.: *A Remote Guidance System for the Blind*. Proceedings of the 12th IEEE International Conference on e-Health Networking, Application and Services 2010, pp. 386-90, 2010.

[17] Arati, K.; Sayali, J.; Sushanta, D.; Harshata, A.: *Object Recognition in Mobile Phone Application for Visually Impaired Users*. IOSR Journal of Computer Engineering vol. 17, pp. 30-33, 2015.

[18] Paeck, T.; Chickering, D.M.: *Improving Command and Control Speech Recognition on Mobile Devices: Using Predictive User Models for Language Modelling*. User Modelling and Useradapted Interaction, pp. 93-117, 2007.

[19] Kowalczyk, M.: *Automatic Face Detection and Identification in Video Stream on the Android Platform*. Bachelor of Engineering Thesis, Lodz University of Technology, 2013.
[20] Li, S.Z.; Jain, A.K.: *Handbook of Face Recognition, Second Edition*. Springer Publishing Company, 2011.
[21] Doukas C.; Maglogiannis, I.: *A Fast Mobile Face Recognition System for Android OS Based on Eigenfaces Decomposition*. IFIP Advances in Information and Communication Technology vol. 339, Springer-Verlag Berlin, pp. 295-302, 2010.
[22] Strahil, R.: *A Fast and Simple Pattern Matching with Hamming Distance on Large Alphabets*. Journal of Computational Biology vol. 23, pp. 874-876, 2016.
[23] AT&T Laboratories Cambridge – The Database of Faces http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

Michał Kowalczyk received in 2015 the M.Sc. degree in the field of computer science at Lodz University of Technology, Faculty of Technical Physics, Computer Science and Applied Mathematics. He continues his works as a PhD. student at the Institute of Information Technology at TUL. The scope of conducted researches is devoted to the subject of image processing with the particular emphasis on mobile systems and stereoscopic materials.

Piotr Napieralski received in 2000 the M.Sc. degree at Lodz University of Technology, Faculty of Technical Physics, Computer Science and Applied Mathematics. He received a PhD degree in 2008 in the field of computer science, computer graphics at the Silesian University of Technology, Faculty of Automatics, Electronics and Informatics. In 2014 he received DSc degree in the discipline of computer science at Lodz University of Technology. He has been working continuously since 2000 at the Institute of Information Technology at TUL. His research interests include: Computer graphics, computer games, virtual reality, user interface design, 3D graphics programming, computer simulation and visualization, computer animation, mobile systems, mobile applications.

Emotion recognition in computer games and films

Filip Rynkiewicz, Piotr Napieralski

Lodz University of Technology, Łódź, Poland
filip.rynkiewicz@dokt.p.lodz.pl

Abstract

In last years technology used in game and film creations has formed a need to check people's reaction to watched images. Human body reacts on external stimulus by face microchanges, distortions in electroencephalography, pupil adjustments etc. Those processes can be recorded with specific apparatus thus correct analysis of those characteristics can be automated. Thanks to this authors are able to check viewer's to their creations, or even construct algorithms that can do it automatically.

Index Terms

emotion recognition, pupil reflex, EEG, electroencephalography, emotion clasificcation

1 Introduction

Studies on recognition of the human emotions can be useful in many areas. Starting with psychology studies on behavioural disorder with patient that have problems with expressing emotions, through biology studies on creation of emotions in human body and ending with getting feedback from a watched movie. Emotions allow to decide if users like what they see or not. That gives them an opportunity to choose if they wants to end it immediately, or even repeat these emotions again. For artists this informations is very desirable, because they can refine their creations based on the information gathered from user's reactions. Thanks to such research artist will know when users will be more interested in action, when it will be more dull or touching for them.

In [1] authors created a theory which explains generation of emotions in human body. They simplify it to few steps, like in a algorithm. First there is a perception of an event then analysis of it based on user's own experience and norms, so finally the event can be classified as certain emotion.

Emotions can be detected by certain characteristics that could be classified as one of the two groups:

- Psychological:
  - EEG(electroencephalography),
  - EMG(electromyography),
  - EKG(electrocardiography),
  - pupil diameter.

- Non-psychological:
  - text,
  - speech,
  - gestures,
  - facial expressions.

This paper will focus on group of psychological signals, especially on EEG and pupil diameter. It will be explained how to detect certain emotion based on fusions of the stimuli. There are plenty of researches where authors combine EEG with pupil diameter or even with eye trackers data, and those combined methods are more reliable and have better accuracy then individual ones [2], [3], [6].

1.1 Subjects and stimuli

Using a variety of movie clips, especially selected for this research and shown to participators, the EEG signal and pupil diameter changes were recorded. The key feature of those movie clips is to cover different emotional responses to get the best and as accurate results. Psychologists recommended videos from 1 to 10 minutes long for elicitation of single emotion [14].

2 EEG

The most popular methods of emotion recognition are based on analysis of electroencephalography signals. Numerous researches [4], [5], [7]–[9] has shown that the brain activity, which EEG collects, is the most reliable source for emotion recognition. The core of those studies is to find brain regions and frequency bands most related to those emotions. Studies of [10] showed that activation for unpleasant emotions was prominent over the right posterior regions in the alpha band. In [11] authors found that frontal brain electrical activity is closely related to musical emotions, and in [12] authors confirmed a theory that gamma band is also related to music emotions.

2.1 Data acquisition

Gathering data of brain activity is done by using special EEG cap, where *AgCl* electrodes placed on it are collecting brain activity in certain areas. Most commonly used layout of electrodes is 10-20 system, shown in Figure 1.

Signals were recorded mostly in 1000–1024 Hz sampling rate. To speed up calculations those characteristics were down sampled to 200–256 Hz. Noises and artefacts reduction were done by applying bypass filter between 0.5 to 70 Hz.

2.2 Data extraction

Correlation of certain spectral power of EEG signal and emotions relevant processing was observed [15]. There are multiple methods of extracting power spectral density (PSD) from raw signals. Two of them will be expounded.

First [6] use Fourier transform and Welch algorithm. This method splits signal into overlapping segments and the PSD is estimated by averaging the periodograms. In result the power spectrum is smoother. PSD of individual electrodes was estimated using 15s long windows with 50 percent overlapping. PSD bands like theta

(4Hz $< f <$ 8Hz), slow alpha (6Hz $< f <$ 10Hz), alpha (8Hz $< f <$ 12Hz), beta (12Hz $< f <$ 30Hz) and gamma 30Hz < f were extracted from electrodes. In additional 14 symmetrical pairs on the right and left hemisphere were extracted to measure possible asymmetry in brain activity.

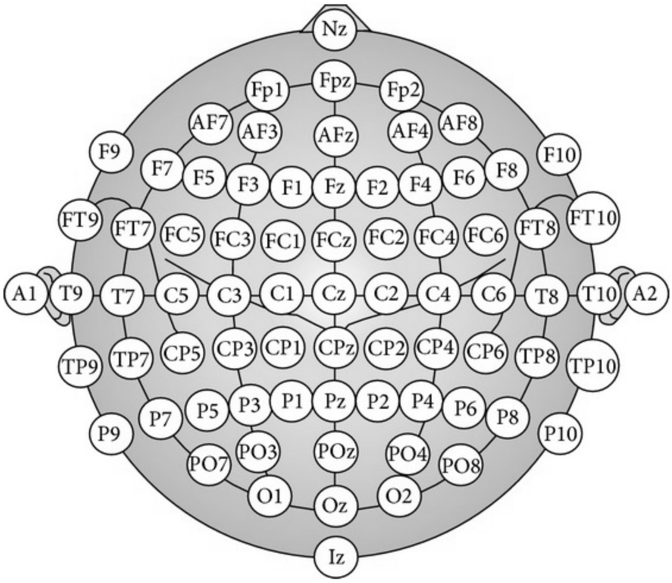


Fig. 1. The EEG cap arrangement for 10-20 system [13]

Second one use a short-time Fourier transform with non-overlapped Hanning window of 4 s. In addition to PSD the differential entropy (DE), differential asymmetry (DASM) and rational asymmetry (RASM) were extracted and compared. Like in first method five frequency bands were used. Delta (1Hz $< f <$ 3Hz), theta (4Hz $< f <$ 7Hz), alpha (8Hz $< f <$ 13Hz), beta (14Hz $< f <$ 30Hz) and gamma (31Hz $< f <$ 50Hz). Using equation 1, DE was calculated.

$$h(X) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} exp\frac{(x-\mu)^2}{2\sigma^2} log\frac{1}{\sqrt{2\pi\sigma^2}} exp\frac{(x-\mu)^2}{2\sigma^2} dx = \frac{1}{2} log 2\pi \mathrm{e} \sigma^2 \tag{1}$$

where X is Gauss distribution $N(\mu, \sigma^2)$, x is a variable π and e are constants. DASM and RASM are defined as:

$$DASM = h(X_{LEFT}) - h(X_{RIGHT}) \tag{2}$$

$$RASM = h(X_{LEFT})/h(X_{RIGHT}) \tag{3}$$

where X_{LEFT} andX_{RIGHT} are DE features of left and right hemisphere of brain.

2.3 Classification

After the data was collected and extracted the support vector machine (SVM) was used as classifier, in both examples. In [6] they smoothed features using linear dynamic system (LDS).

TABLE 1
The performance of classifiers in % using different kinds of frequency band features. For [6]

Feature		Frequency Bands					
		Delta	Theta	Alpha	Beta	Gamma	Total
PSD	Mean	51.60	51.87	54.74	23.23	51.36	**59.04**
	Std	19.56	14.48	16.58	18.06	16.10	20.31
DE	Mean	70.51	47.98	60.17	64.29	68.73	**71.77**
	Std	12.18	15.19	12.94	23.05	20.30	12.03
DASM	Mean	61.08	43.42	49.98	46.96	64.12	**68.37**
	Std	22.45	19.45	15.59	15.21	22.94	23.86
RASM	Mean	61.44	44.90	48.69	48.18	62.71	**66.03**
	Std	22.90	12.14	16.68	15.93	21.11	24.62
ASM	Mean	65.18	44.78	50.29	45.19	63.92	**67.91**
	Std	22.32	13.87	15.91	12.77	22.19	24.45

Result of classification can be bee seen in the Table 1. ASM feature is concatenation of DASM nad RASM. As we can see, delta and gamma frequency bands perform better than theta and alpha frequency bands, and total frequency band has a stable and prominent accuracy. Also we can see that, differential entropy features get best accuracies in almost all frequency bands except Theta band (47.98% of DE features is less than 51.87% of PSD features).

In [3] in EEG there was only DE feature. They have used SVM classification with RBF kernel. Result of classification can be seen at Table 2.

TABLE 2
The performance of classifiers using different kinds of frequency band features. For [3]

Arousal classification			**Valance classification**		
Band	Electrode/s	$\sigma^2_{bw}/\sigma^2_{wn}$	Band	Electrode/s	$\sigma^2_{bw}/\sigma^2_{wn}$
Slow α	PO4	0.18	β	T8	0.08
α	PO4	0.17	γ	T8	0.08
θ	PO4	0.16	β	T7	0.07
Slow α	PO3	0.15	γ	T7	0.06
θ	Oz	0.14	γ	P8	0.05
Slow α	O2	0.14	γ	P7	0.05
Slow α	Oz	0.14	θ	Fp1	0.04
θ	O2	0.13	β	CP6	0.04
θ	FC6	0.13	β	P8	0.04
α	PO3	0.13	β	P7	0.04

The linear discrimination criterion was calculated for EEG signals. Dividing between class variance by within class variance for any given feature. For arousal classification, PSD in alpha bands of occipital electrodes was found to have the most discriminant features. In contrast, valence beta and gamma bands of temporal electrodes are more informative. The between class to within class variance ratios are higher for the best arousal EEG features. The higher linear discrimination criterion for best arousal features explains the superior classification rate for arousal dimension.

3 PUPIL DIAMETER

The theory was forged based on the observation of the human eye. Pupil diameter is changing in different emotional states. The disadvantage of this solution is that pupil's diameter is highly dependant on the light. First step to gathering data from a pupil is to create lighting reflex model. The most common and the simplest method is principal component analysis (PCA).

3.1 Light reflex model

Assuming that Y is the $M \times N_p$ matrix of pupillary response for the same picture for N_p participant and M samples, Y consists of three components

$$Y = A + B + C \tag{4}$$

where A is the lighting influence, B is the emotional response and C is the noise. Extracting first component from PCA to approximate the pupil response for the lightning changes during the experiments.

3.2 Data acquisition

To gather the data the Eye-Tracker devices were used. These apparatus are able to collect position of the projected eye gaze on the screen, pupil diameter, moments when the eyes were closed and a distance of the participant's eyes to the gaze tracker device. Eye blinking creates gaps in eye gaze and pupil record, thus the linear interpolation was used to replace missing samples.

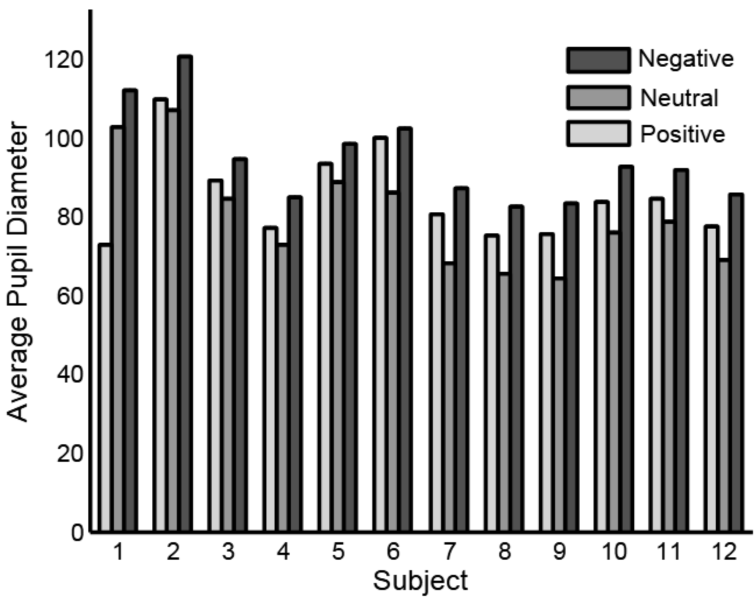


Fig. 2. Average pupil diameter. For [6]

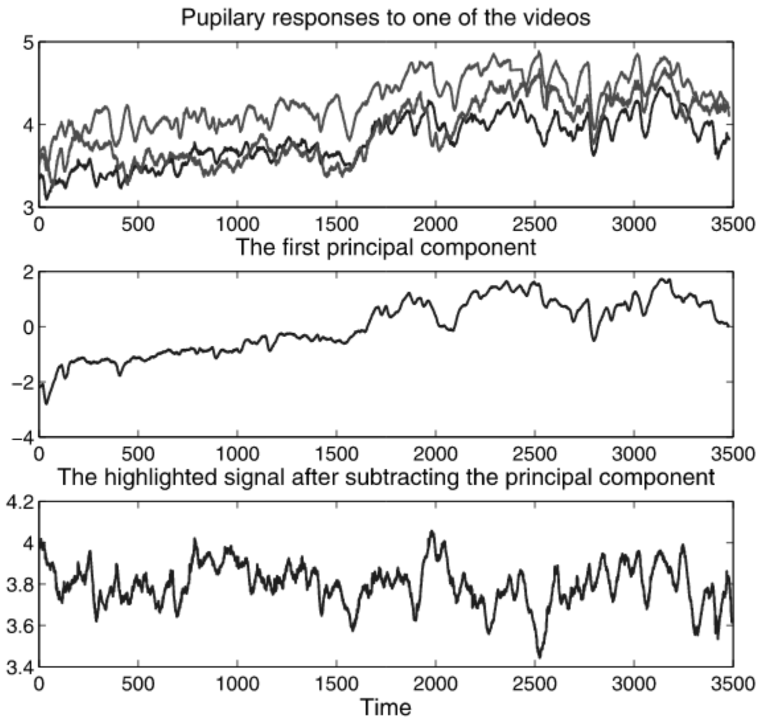


Fig. 3. From top to bottom: In the first plot, there is an example of pupil diameter measures from three different participants in response to one video. The second plot shows the first principal component extracted by PCA from the time series shown in the first plot (the lighting effect). The bottom plot shows the pupil diameter of the blue signal in the first plot after reducing the lighting effect [3]

As it can be seen at Figure 2 for 12 participants the smallest average value have neutral emotion, except subject 1.

In the Figure 3 the examples of pupillary responses, extracted pupillary lighting reflex, and the residual component after removing the light reflex are given.

3.3 Classification

As in the EEG for classification of signal, for both examples, the SVM was used.

As it can be seen at Table 3 the DE feature performs much better than PSD, because DE features have the balance ability of discriminating patterns between lowand high frequency energy.

4 Multimodial Fusion

After the data was gathered the fusion of methods is what's left. Examples used in this article have implemented the feature level fusion and decision level fusion. First mentioned approach use the features vectors from different stimuli and concatenate them to form larger feature vector. Second one use two classifiers which were trained with different features, and fused to generate a new classification using some new

TABLE 3
Performance in % of using different features from pupil diameter [6]

Exp	Feature	Acccuracy	Exp	Feature	Accuracy
1	PSD	65.43	7	PSD	33.95
	DE	86.42		DE	**61.73**
2	PSD	56.79	8	PSD	46.91
	DE	**70.37**		DE	**50.63**
3	PSD	54.94	9	PSD	43.83
	DE	**56.79**		DE	**59.88**
4	PSD	60.49	10	PSD	36.42
	DE	**63.58**		DE	**59.88**
5	PSD	37.65	11	PSD	44.44
	DE	**4.877**		DE	**50.62**
6	PSD	33.95	12	PSD	34.57
	DE	**4753**		DE	**5062**
Mean	PSD	45.78	Std	PSD	11.03
	DE	**58.90**		DE	**10.25**

principles or learning algorithms. In [6] they applied two principles. One was called max strategy which selected the higher probabilistic outputs of classifiers trained with a single modality separately as final result. Another was called sum strategy which summed up probabilities of same emotions from different frequency bands and selected the highest one.

4.1 Results

Overall results of experiments for [6] are shown at Table 4. We see that decision level fusion using max strategy and feature level fusion performed better than single modality like EEG or pupil diameter, which achieved average accuracies of 72.98% and 73.59%, respectively.

TABLE 4
Performance in % of using different multimodal features [6]

	EEG (DE)	Max Strategy	Sum Strategy	Feature Fusion
1	83.09	83.09	83.09	**93.59**
2	68.22	68.22	51.31	**78.72**
3	68.22	67.93	51.02	**68.22**
4	85.13	68.22	**85.13**	83.97
5	51.31	51.31	51.31	**77.55**
6	83.09	**83.09**	**83.09**	**83.09**
7	51.31	**68.22**	**68.22**	58.02
8	83.09	**83.09**	83.09	**83.38**
9	68.22	83.09	68.22	63.56
10	68.22	68.22	68.22	**69.10**
11	**68.22**	**68.22**	**68.22**	40.82
12	**83.09**	**83.09**	65.89	**83.09**
Mean	71.77	72.98	68.90	**73.59**
Std	12.03	**10.09**	12.85	14.43

Another results Table 5 for [3] has shown that the DLF have the best accuracy for arousal and valance.

TABLE 5
Performance in % of using different multimodal features [3]

Modality	Classification rate	
dimension	arousal	valance
EEG	62.1%	50.5%
Eye gaze	71.1%	66.6%
Feature level fusion (FLF)	66.4%	58.4%
Decision level fusion (DLF)	**76.4%**	**68.5%**

Difference of results of both researches are really high. For FLF the percentage is accordingly 66.4% and 73.59%. EEG for PSD feature 62.1% and 71.77% for DE feature.

5 Conclusions

Emotions are sophisticated mechanism in human body, but knowledge how they work can be helpful in many areas. Those signals can be obtained from many of human impulses, such as pupil reflex or brain signals. Using appropriate techniques and devices those characteristics can be collected and analysed to detect emotions. Combining those modalities and fusing it with special techniques can lead to better accuracy of methods.

References

[1] Ortony, A.; Clore, G.L.; Collins, A.: *The Cognitive Structure of Emotions.*. Cambridge University Press, July 1988.

[2] Calvo, R. A.; D'Mello, S.: *Affect detection: An interdisciplinary review of models, methods and their applications*. IEEEE Transactions on Affective Computing, vol 1, no1, pp 18-37, 2010.

[3] Soleymani, M.; Pantic, M.; Pun, T.: *Multimodal Emotion Recognition in Responce to Videos*. IEEE Transaction on Affective Computing, vol. 3, no. 2, April-June 2012.

[4] Adolphs, R.; Tranel, D.; Damasio, A.R.: *Dissociable Neural Systems for Recognizing Emotions*. Brain and Cognition, vol. 52, no. 1, pp. 61-69, June 2003.

[5] Damasio, A.R.; Grabowski, T.J.; Bechara, A.; Damasio, H.; Ponto, L.L.B.; Parvizi, J.; Hichwa, R.D.: *Subcortical and Cortical Brain Activity during the Feeling of Self-Generated Emotions*. Nature Neuroscience, vol. 3, no. 10, pp. 1049-1056, Oct. 2000.

[6] Wei-Long, Z.; Bo-Nan, D.; Bao-Liang, L.: *Multimodal Emotion Recognition using EEG and Eye Tracking Data*. IEEE, 2014.

[7] Lin, Y.P.: *EEG-Based Emotion Recognition in Music Listening*. IEEE Transactions on Biomedical Engineering, vol. 57, no. 7, pp. 1798-1806, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5458075&isnumber=5484937, 2010.

[8] Gao, Y.; Lee, H.J.; Mehmood, R. M.: *Deep learninig of EEG signals for emotion recognition*. IEEE, International Conference on Multimedia & Expo Workshops (ICMEW), Turin, pp. 1-5, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7169796&isnumber=7169738, 2015.

[9] Nie, D.; Wang, X.-W.; Shi, L.-C.; Lu, B.-L.: *EEG-based Emotion Recognition during Watching Movies*. IEEE, Proceedings of the 5th International IEEE EMBS Conference on Neural Engineering Cancun, Mexico, April 27 – May 1, https://pdfs.semanticscholar.org/6511/590bc9677922c82747b5d183383f46b50db6.pdf, 2011.

[10] Sarlo, M.; Buodo, G.; Poli, S.; Palomba, D.: *Changes in EEG alpha power to different disgust elicitors: the specificity of mutilations*. Neuroscience Letters, vol. 382, no.3, pp. 291-296, 2005.

[11] Schmidt, L.A.; Trainor, L.J.: *Frontal brain electrical activity distinguishes valence and intensity of musical emotions*. Cognition and Emotion, vol. 15, no. 4, pp. 487-500, 2001.

[12] Li, M.; Lu, B.L.: *Emotion classification based on gamma-band EEG*. IEEE Int. Conf. Engineering in Medicine and Biology Society, Minneapolis, pp. 1223-1226, 2009.

[13] Jirayucharoensak, S.; Pan-Ngum, S.; Israsena, P.: *EEG-Based Emotion Recognition Using Deep Learning Network with Principal Component Based Covariate Shift Adaptation*. The Scientific World Journal, 2014.

[14] Schaefer, A.; Nils, F.; Sanchez, X.; Philippot, P.: *Assessing the effectiveness of a large database of emotion-eliciting films: A new tool for emotion researchers*. 2010.
[15] Aftanas, L.I.; Savotina, L.N.; Makhnev, V.P.; Reva, N.V.: *Analysis of evoked EEG synchronization and desynchronization during perception of emotiogenic stimuli: association with autonomic activation processes*. Neuroscience and Behavioral Physiology. 35(9), pp. 951-957, 2005.

Filip Rynkiewicz received his M.S degree in Computer Science from Technical University of Lodz in 2016. Currently working at his PHD in decoupling light in case of human vision. As the developer working in Ericsson, engaged at the telecommunications area using cutting edge technology in LTE. In free time developing knowledge in template metaprogramming, playing guitar and giving his all efforts as the beginner thriatlete

Piotr Napieralski received in 2000 the M.Sc. degree at Lodz University of Technology, Faculty of Technical Physics, Computer Science and Applied Mathematics. He received a PhD degree in 2008 in the field of computer science, computer graphics at the Silesian University of Technology, Faculty of Automatics, Electronics and Informatics. In 2014 he received DSc degree in the discipline of computer science at Lodz University of Technology. He has been working continuously since 2000 at the Institute of Information Technology at TUL. His research interests include: Computer graphics, computer games, virtual reality, user interface design, 3D graphics programming, computer simulation and visualization, computer animation, mobile systems, mobile applications.

STERIO – Reconstruction of 3D Scenery for Video Games Using Stereo-photogrammetry

Mariusz Szwoch*, Adam Kaczmarek†, Dariusz Bartoszewski‡

* Gdańsk University of Technology, Gdańsk, Poland, szwoch@eti.pg.edu.pl
† Gdańsk University of Technology, Gdańsk, Poland, adam.l.kaczmarek@eti.pg.edu.pl
‡ Forever Entertainment S.A., Gdynia, Poland
dariusz.bartoszewski@forever-entertainment.com

Abstract

In this paper, the main assumptions of the STERIO project are presented, which main goal is to develop a complete technology of faithful reconstruction of the real world sceneries in a virtual world of video games using stereo-photogrammetry methods. The proposed approach should be cheaper than competitive laser scanning, and at the same time faster and more accurate than existing methods based on photogrammetry. Such technology will significantly speed-up at a reasonable costs the process of construction of 3D scenery based on locations from real world. Thus, the presented solution should be interesting for many small independent game developers.

Index Terms

stereo-photogrammetry, 3D scanning, video games

1 Introduction

Modern 3D video games require hard use of high quality 3D assets for both game characters as well as level scenery. In the traditional approach, all such assets are created by 3D artists or bought from a store. The first approach is a very labor-intensive task while the second one, does not guarantee originality of the assets, which is easily recognized by players. Situation becomes more complicated in case when game concept requires that action took place in a location that represents a real world one or similar to real locations. In such a case, handmade modelling seems to be non-creative yet time and labor consuming task that could be replaced by automatic or at least semi-automatic machine work. This was not a big concern when 3D graphics complexity calculated in real time in gaming industry was very limited by CPU and GPU units' strength. Main goal for 3D artists was intelligent simplification of the scene or even to trick human minds to see more than it was really shown. However, nowadays players expect photorealistic graphics quality, similar to 3D scans.

The video game development market is evolving at a very dynamic pace what forces programmers all over the world to keep adapting to new systems, platforms and user preferences, and continue improving existing programming. For this reason, development studios require new, better, and most importantly faster methods of handling such issues as real world digitization, which is based on converting real places into the digital ones and implementing them in 3D game engines.

If limited to 3D scanning of inanimate objects, there are several possible solutions available including methods based on lasers, structured light RGB-D sensors, and photogrammetry [1], [2]. Unfortunately, all of the mentioned solutions have some drawbacks that limit their usage in video game industry, especially by small and medium developer teams. Handmade 3D modelling requires a lot of human labor that increase with the model complexity. Laser based scanners are very precise as to geometry scanning but are very expensive and does not support textured models. RGB-D scanners, based mostly on the structured infrared light patterns, like Microsoft Kinect or Intel RealSense, provide complete 3D models of usually enough quality but are limited to small distances of about several meters [3], [4]. Finally, photogrammetry methods can provide high quality textured models but require numerous input data and high computational costs.

In this paper, the main concept of the STERIO project is presented, which aims to overcome problems of above approaches by using the stereo-photogrammetry methods, which, under some assumptions, can offer reliable scanning of textured objects at reasonable financial and computational costs.

2 Background

The video game sector is one of the most dynamically developing fields of entertainment. Newzoo's analyses show that in 2016, its global revenue reached USD 99.6 and at the current growth pace (6.6% per year), the global value of the game market could reach USD 118.6 billion by 2019 [5].

Such a big market is created both by well-recognized big development studios with a huge budget for AAA productions as well as by small independent development studios (*indies*), enthusiasts, and creative teams that often have insufficient production resources and are limited by a shortage of professional, top-standard technology. The small development studios often have put years of work into creation of a quality product sufficiently developed for commercialization purposes.

Nowadays, most video games are developed using game engines but only few big development studios are able to develop their own ones. Most video games developers use one of a few commercially available game engines such as Unity or Unreal Engine. There is no available any reliable data about financial share of each engine on the market, but with no doubt Unity 3D is the leader engine in the small- and medium-sized developer studios. According to market Statista, in 2012 62% of game developers in UK declare that they use Unity 3D [6]. According to Unity enterprise, Unity 3D is currently the most popular game engine, with the total number of registered developers reaching roughly 4.5 million [7]. Even, if only a small part of them are active in professional game development and distribution, it gives a great number of smaller studios (with up to around a dozen members) developing action, adventure, shooter, and/or FPS games, which should be interested in purchasing modern but low-cost 3D technologies and solutions.

Market dynamics force teams of programmers to keep adapting to new player expectations and preferences. This usually entails the need for the newest, most expensive technologies and/or higher production costs to speed up and improve the developed game. It is also the reason why video game development studios require new, better and, most importantly, faster methods to handle problems appearing in the development process. One such problem is the effective and precise recreation of real places in the digital world. Currently, a game developer has two options to recreate a room, as an example, and define the parameters of the objects within it in 3D graphics.

The first, is to take photos of the room with a wide-angle lens, merge them into one frame with dedicated software, and manually calculate the size of specific objects, their shapes, the location in the room, and relative positioning of other objects, to use as the basis for arranging the objects and importing them into the graphical module of the 3D engine.

The second solution is to apply 3D laser-scanning technology, which provides a very precise and detailed image. Depending on individual preferences, numerous scanners may have to be used simultaneously. Using this technology, the scanned room is uploaded into a special program for processing the images created by the laser scanner and used as the basis for creating the room's map. This form of digitization eliminates the need for manual calculations and the recording of specific parameters as that data is collected during the 3D laser-scanning stage.

The first of the aforementioned solutions entails lower financial outlays for dedicated equipment, but is very labor intensive on the side of the employees. This method for room digitization is used mainly by small development studios that cannot afford the second option (laser-scanning technology), which requires not only high capital costs (professional scanners manufactured by such companies as Artec or Creaform can cost as much as $50.000), but also competent operators and professional, expensive software adapted for processing the images produced by the scanners. When game developers choose this method, it always considerably simplifies the recreated world and produces an inferior visual effect, which is why it is usually applied in low-budget productions. This is the reason, why we are attempting to create an innovative method using and integrating stereoscopic photography technology (stereo-photogrammetry) into the Unity 3D game engine.

In the next section, the main assumptions of the STERIO project are presented. Then, results of some initial experiments are reported. Finally, some conclusions and future works are given.

3 The STERIO project

The STERIO project aims at the development of technology and a subsequent prototype of a programming tool for the effective development of 3D graphics using stereoscopic photography [8], [9]. The result of the project will allow for faithful recreation of real world scenery in the virtual world of video games. Applying the proposed technology will allow for more effective and less costly location generation for games in real places. The productions developed with this technology will provide the impression of movement in the real world, which will improve the emotional experience of the players and generate demand for the proposed solution. This method will have a big advantage over the indirectly competitive 3D laser-scanning technology as it produces a similar effect with fewer resources and shorter production time, which can reach even up to 40%. The savings come from using a stereoscopic unit composed of a system of two standard cameras (semi-professional, single lens reflex cameras). To facilitate operation, the technology will first be integrated with the most popular Unity 3D engine.

The result will be a technology (including methods, algorithms, prototype of the programming tool, and plug-in for the Unity 3D engine) that supports video game production, i.e. 3D graphics generation through the conversion of stereoscopic images into 3D objects subsequently used by the game engine. To achieve the aforementioned goal, the project will be carried out in four stages:

1) development of optimal stereoscopic image production and configuration of hardware for reference purposes;
2) development of algorithms for converting stereoscopic data into properly defined 3D objects;
3) development of a prototype tool for video game producers to be used as a plug-in for the Unity 3D;
4) development of a reference game for experimental verification of the result and final proof of its effectiveness.

The main purpose of the proposed solution is to support and improve the video game development process with an original method to recreate reality in a digital world. In this project, both the method and technology dedicated to the Unity 3D engine will be developed to improve rendering of 3D space in video games. This technology will be an application, a plug-in for the Unity 3D engine, used to import and analyze stereoscopic photos for point clouds and convert them into 3D objects. The proposed tool and method will support the development of video games for various platforms and devices. At the first attempt, the technology will be dedicated to the Unity 3D engine, the universal nature of which allows for the development of games for most of the currently available platforms and devices. This innovation in the process and technology of 3D object digitization will serve as a somewhat competitive solution for the considerably more expensive laser scanning technology and can be used for purposes other than video game development, e.g. architectural visualizations, simulations and other. This approach will constitute creative technology for the benefit of image production development and automation, as it is a technique for digitizing and processing multidimensional objects and consequently improving the development of creating 3D objects from the ground up.

Because the STERIO project focuses on the reconstruction and modelling of real places in the digital world, the results will be important mostly to developers of 3D FPS (*first person shooters*) or other action games. However, the advantages of the technology, especially the little time required to develop 3D graphics, will make most teams deem such a tool necessary. STERIO is directed mainly at teams, which:

- Prioritize a faithful reconstruction of real scenes.
- Specialize in low-budget productions and release many games commercially in production cycles that are as short as possible.
- Demand low-cost technology to considerably accelerate their development work, yet still simultaneously provide the user with a high experience level.
- Reject purchasing the expensive and labor-intensive 3D laser-scanning technology as it does not meet their needs or exceeds their investment potential.

- See a different value in the proposed technology, e.g. potential for its evolution to adapt it to their own needs or to be combined with a different internal tool, etc.

It is also important that the proposed technology relies on the development of a plugin for Unity 3D – the most popular and universal video game engine. It should be noted that the STERIO technology can be successfully integrated with other game engines, but Unity 3D was chosen for the purposes of this project because of the project's commercialization strategy.

4 The Proposed Approach

The technology developed in the STERIO project will be, generally, based on merging 3D images acquired with a pair of cameras fixed beside each other on a camera rig [10]. The cameras may be of a different quality with regard to the game development budget, both professional and amateur equipment can be used. Therefore, the technology will be suitable for low-cost game development.

The technology is designed for making 3D models of interiors of existing buildings, which can be used in practically any genre of video games. Developers in order to acquire a 3D model of interiors of the building will have to make a series of images of these spaces. Images need to be taken from different points and angles, and determining the sub-optimal way of making these photos will be one of the tasks within the STERIO project. Each stereo-image will provide a 3D image of a part of the scene. The application created within the project will process input images and generate a 3D model of a whole scene.

In order to improve the quality of 3D images stereo cameras have to be calibrated. The calibration allows to avoid distortions of different kinds (barrel, pincushion, fish-eye, e.t.c) and, above all to minimize the depth restoration error. The calibration performed in the STERIO project will be based on taking a series of images of a precisely defined calibration chessboard pattern. The developed application will analyze the distortion created by cameras and it will reduce them by transforming images.

The technology developed in the STERIO project will be based on stereo-cameras. There are also other technologies, which create a 3D model based on a series of images taken from different points of view with the use of a single camera. However, using stereo cameras instead of a single one will result in better quality of 3D models. It will also make possible to better processing of difficult scenes, with uneven illumination, or containing flat homogeneous surfaces with no salient point. The following chapters describe some initial experiments performed to verify the presented assumptions.

5 Preliminary Results

The preliminary research and experiments performed in the STERIO project verified the capabilities of existing commercial applications and open source libraries designed for acquiring 3D models from images. The experiments were performed with two commercial applications: Autodesk Remake (https://remake.autodesk.com/about) and Agisoft Photoscan (http://www.agisoft.com/). These applications were used to obtain 3D models based on a set of images taken by a single camera from different points of view. Additional experiments verified efficiency and accuracy of two open source libraries: openMVG and openMVS for the reconstruction of spare and dense point clouds based on testing dataset.

The commercial software bases its results on a series of images from a single camera and there are no common commercial applications for obtaining a 3D model on the basis of pairs of images from a stereo camera. However, it is possible to regard images from a stereo camera as two images taken by the same camera from two different viewpoints. In such approach, images from stereo cameras can be used as input data. In order to perform reliable experiments a special image data sets have been created from different test locations.

5.1 The STERIO data set

The first stage of the STERIO project assumes preparing a comprehensive data set of images allowing for further research in different aspects of stereo-photogrammetry based interiors scanning for video games application. These aspects include research on proper camera settings and processing of difficult surfaces such as transparent, semi-transparent and light-reflexing materials, flat surfaces with low number of salient points etc. The whole dataset will consist of several test locations and the final one that will be used for validation of developed algorithms and scanned as a main location in the test video game. For each location, a different number of data subsets will be prepared consisting of images taken under different lighting conditions with different exposition parameters, camera location and orientation points, additional virtual markers displayed, and other experiment setting. Each dataset will be accompanied with additional images of a special chessboard pattern for calibration of a camera stereo-pair.

The first data set used in the experiment consisted of about 2500 photos of interiors of a baroque manor house in southern Poland. This residence is a good location for any adventure video games thanks to its compact but varied body and rooms full of different details but without furniture (Fig. 1). The data set contained images of antique chambers, corridors and stairs. All images were taken from different points of view without a tripod. Generally, photographs were made around the perimeter of each room at a distance of about 3m with overlap of about 40%. Moreover, additional photos were made of complicated, decoration elements like sculptures, grilles, handrails etc. All photos in this subset were taken with automatic settings using a standard camera.

5.2 Results provided by Autodesk Remake

In the initial experiments a trial version of Autodesk Remake was used, which allows to process the set of no more than 50 images. In order to verify the results of Remake, the data set was divided into parts. Each part consists of images of the same interior. A sample result provided by Autodesk Remake is presented in Fig. 2a. The figure presents the part of a of the room 2 shown in Fig. 1b with the wireframe of the triangle mesh geometry visualized.

Autodesk Remake was able to restore the whole shape of the room based on limited number of images. The number of triangles included in the resulting 3D model was below 1 000 000 for a half of a room. The model with a significantly high number of triangles in the mesh cannot be used as a map in a 3D game because it needs to be instantly processed by a graphic card. The most desired result is such

Fig. 1. Sample images of rooms in a manor house from southern Poland

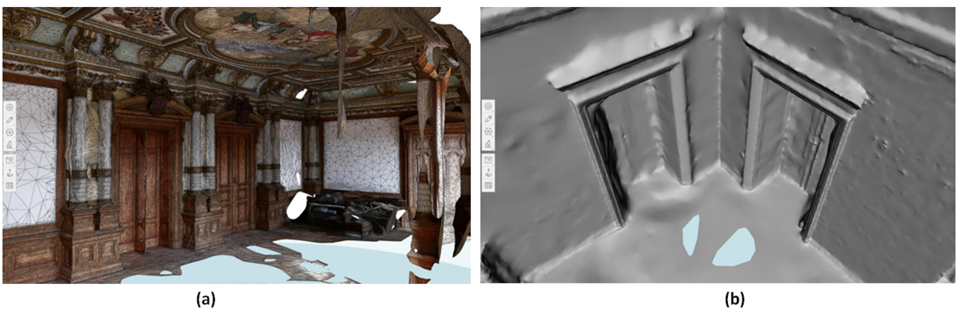


Fig. 2. A sample model of a room acquired by Autodesk Remake: a textured mesh with a wireframe (a), a mesh showing different kinds of geometrical deformations (b)

that there is a low number of triangles covering simple surfaces like walls and the number of triangles is high in the areas with a complicated shape. Autodesk Remake generated the result, which partly meet these requirements.

Although, in general the model acquired by Autodesk Remake is correct it contains many undesirable features. In particular, complicated shapes of some decoration elements were simplified. Moreover, some objects are deformed and walls are not perfectly flat. These effects are visible in Fig. 2b, which shows a part of shape of a model without textures.

(a)

(b)

Fig. 3. Sample results from Agisoft Photoscan: a textured fragment of a room (a), geometry distortions on flat surfaces in a corridor (b)

5.3 Results provided by Agisoft Photoscan

Agisoft Photoscan does not limit the number of input images, so the entire data subset was used in the experiment. In general, the software correctly created the 3D model (Fig. 3a), however there were areas in which the application generated unacceptable results. For example, Photoscan was not able to correctly acquire the model presented in Fig. 3b.

Generally, well-lit surfaces were significantly better reproduced than dark ones. Photoscan created a triangle mesh containing few millions of triangles, which has the same density in both flat (simple) and complicated areas. This number is too high for the purposes of game development. The application includes also algorithms for simplifying the triangle mesh. Unfortunately, the geometry reduction is performed in all areas, regardless of their shape. The number of triangles becomes lower both in simple walls as well as in detailed decorative elements, which obviously is not a desired effect.

5.4 Tests with openMVG library

The openMVG (*multiple view geometry*) is an open source library for computer-vision scientists and especially targeted to the Multiple View Geometry (MVG) problems [11]. It provides many useful functions, in particular, calculation of camera positions based on a series of images and generation of a rare point cloud representing photographed object. To achieve this, the library uses two algorithms:

- Incremental Structure from Motion (ISfM), which involves iterative addition of new views and 3D points using position estimation and triangulation.
- Global Structure from Motion (GSfM), which estimates the position and orientation of images in 3D using the global calibration method, based on the fusion of relative displacements between pairs of images.

To test the library, its source files were prepared for compilation using CMake, and then compiled in Microsoft Visual Studio environment. In the first experiment, a series of photographs of the Fahrenheit Courtyard at Gdańsk University of Technology (GUT) were processed. The test scene is characterized by a rather complex geometry with numerous refractions, uneven illumination with shaded surfaces, transparent windows and a large flat surface of a whiteboard without any salient features (Fig. 4). After tuning the algorithm's parameters, the openMVG library allowed to properly estimate positions of the cameras and to generate a rare cloud of 68,278 points (Fig. 5a). Although, the result is quite satisfactory, especially as regards the reconstruction of window openings, it is easy to see the limitations of the library in reconstructing of flat surfaces without characteristic points.

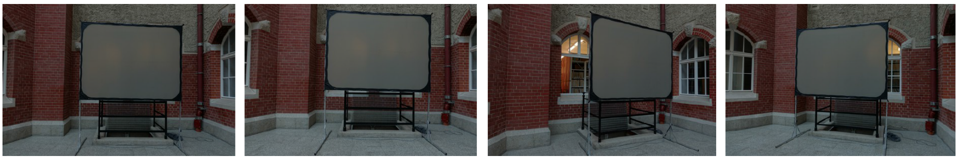
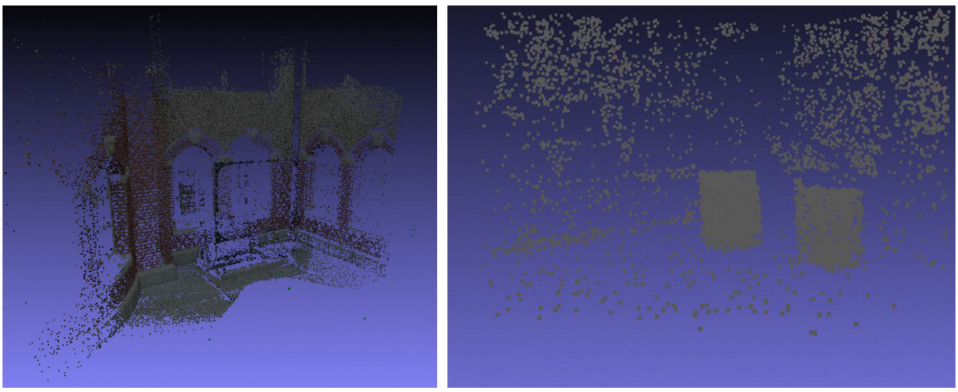


Fig. 4. Sample fotographs of the Fahrenheit Courtyard at Gdańsk University of Technology



(a) (b)

Fig. 5. Rare point clouds received from openMVG library: from series of photographs from Fahrenheit Courtyard (a), from a stereo-pair of a test room corner (b)

Dealing with this type of problems is one of the main objectives of the STERIO project. One of the possible solutions for reconstruction of such uniform flat surfaces is using an LCD projector displaying a special pattern on them. In another test, a test room corner with uniform light grey walls and a blackboard table, illuminated by a pattern from a multimedia LCD projector, was photographed with a stereo-rig of two cameras. Only a single stereo-pair (left and right images) was processed by openMVG library allowing to reliably reproduce the point cloud of the flat surfaces, that was impossible in the previous test (Fig. 5b). In this case, the openMVG library was used with the "-p HIGH" parameter, which caused searching for specific points with greater accuracy.

5.5 Tests with openMVS library

The openMVS (*multi-view stereo*) is an open source library that can be used to reconstruct an object surface based on a series of object's images and camera positions and a rare points cloud, which can be generated for example by the openMVG library. The resulting texture model mesh is created as a result of the following reconstruction steps:

- Dense point-cloud reconstruction based on the algorithm "Patch-Match" (A Randomized Correspondence Algorithm for Structural Image Editing).
- Mesh reconstruction for estimating a mesh surface that best fits the given point-cloud.
- Mesh refinement to increases the detail of the model mesh.
- Mesh texturing based on original high-resolution photos.

This library source files were also prepared for compilation using CMake, and then compiled in Microsoft Visual Studio environment. The resulting application allowed to continue both experiments from the previous section. The resulting files created by openMVG were exported to the binary files in the format accepted by openMVS. For the first scene of the Fahrenheit Courtyard we received: a dense cloud of over 1.8 million points (Fig. 6a), a model mesh with over 1.3 million vertices (Fig. 6b), a detailed mesh of model simplified with about 650 thousand vertices (Fig. 6c), and a mesh model with textures (Fig. 6d).

The final effect is very visually pleasing, and the only obvious shortcoming is the empty surface of the whiteboard. However, in-depth study of the model makes it possible to identify a few additional problems, including some geometry deformations resulting from the reconstruction errors, and too complex object mesh for the practical use in video games. These problems are also evident in the final result of the second scene (the room corner), though here it has been possible to reproduce flat homogeneous surfaces correctly by using an LCD projector (Fig. 7a). All these two problems should be addressed in the STERIO project.

5.6 Using models in Unity 3D

The last experiments was to import the triangle meshes generated by Autodesk Remake, Agisoft Photoscan and a prototype application using openMVG and openMVS libraries into the Unity 3D environment used in the game development process. All models, regardless of their source, were successfully imported to Unity 3D by using the fbx file format. A sample scene from the Unity 3D is presented in Fig. 7b.

6 Conclusions and Future Work

In this paper, the concept of using stereo-photogrammetry methods to perform reconstruction of the real world sceneries for creation of virtual world in video games was presented. The proposed approach can significantly reduce the time required to develop an FPS game approximately by 20-40%. Since the main costs of game development are the labor costs for programmers and 3D graphics engineers, this will also entail roughly 20-40% savings in the total production costs. Furthermore, reducing production time will allow for the development of more games in the same time, thus increasing potential revenue. Thanks to the planned results of the project,

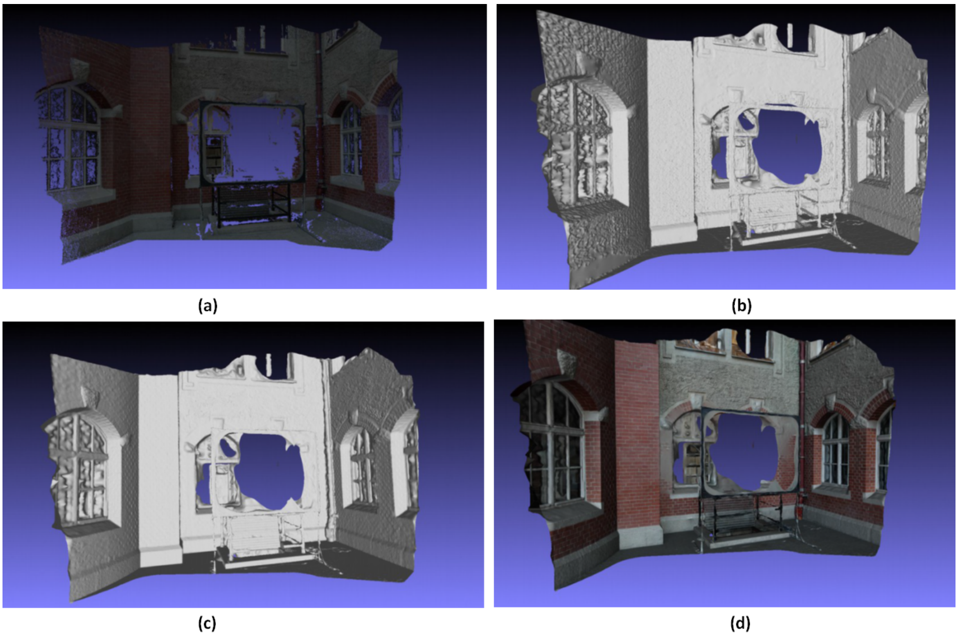


Fig. 6. Results of the subsequent steps of the model reconstruction process using openMVS library: a dense point cloud (a), a model mesh (b), a detailed model mesh (c), and complete model mesh with a texture (d)

Fig. 7. A sample reconstruction of a flat surface illuminated by a graphical pattern from an LCD projector (a), a sample reconstructed scene imported into Unity 3D environment (b)

the development potential of small or medium studios will increase due to receiving a comprehensive tool for faithful reproduction of reality in video games.

Thanks to STERIO, development studios will have the opportunity to recreate places from the real world in the digital world in a few simple steps. The first is to take properly stereoscopic photos of a specific room with the stereoscopic imaging system. The next is to use the application (plug-in) to import and analyze the stereoscopic photos to create a 3D map of the room, which will include all parameters necessary to provide the user with the best possible experience – mainly the shape and arrangement of individual objects inside the room in an absolute sense and with respect to each other. Digitization with the proposed technology will allow development studios to minimize the time necessary to develop a video game, consequentially reducing associated costs. Furthermore, this solution will make the places recreated in the digital world looking authentic and will provide the player with a more realistic experience, thus contributing to the commercial success of the given production.

The performed initial experiments confirmed that faithful architectural restoration is possible by the means of photogrammetry. Unfortunately, this approach has also some disadvantages including high computational complexity and significant problems with a point cloud construction in case of lacking salient points in paired images. By proposing the use of stereo-photogrammetry, we hope to overcome these limitations due to earlier, more reliable and precise reconstruction of point clouds.

The development of STERIO project may lead to a new trend in gaming industry. The project makes it possible to acquire at low cost a 3D map resembling a real building. Therefore, it is possible to popularize games with these kinds of sceneries. The action of the game can take place in a building, which is known to a player. Moreover, 3D maps reflecting real buildings can also be prepared to existing games. Therefore, game development studios can prepare and sell these kinds of maps.

Our future work within the project will focus on several problems including: point clouds' matching, and reliable complexity reduction of the reconstructed mesh. Although, at the first stage STERIO tools will be compatible with Unity environment, the technology will be also integrated with other game engines such as Unreal Engine etc.

ACKNOWLEDGMENT

This work was supported by the Sectoral Programme GAMEINN within the Operational Programme Smart Growth 2014-2020 under the contract no POIR.01.02.00-00-0140/16.

REFERENCES

[1] Ahmadabadian, A.H. *et al.*: *A comparison of dense matching algorithms for scaled surface reconstruction using stereo camera rigs*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 78, 157–167, 2013.

[2] Kaczmarek, A.L.: *Improving depth maps of plants by using a set of five cameras*. Journal of Electronic Imaging, vol. 24, no. 2, 023018, 2015.

[3] Szwoch, M.; Pieniazek, P.: *Detection of Face Position and Orientation Using Depth Data*. The 8th Intern. Conf. on Image Processing and Communications; Image Processing And Communications Challenges 7 Book Series: Advances in Intelligent Systems and Computing Volume: 389 Pages: 239-251; 2016.

[4] Szwoch, M.; Pieniazek, P.: *Facial Emotion Recognition Using Depth Data*. The 8th Int. Conf. on Human System Interaction, 271–277, IEEE, 2015.
[5] Newzoo Games *Global Games Market Report 2016*. https://cdn2.hubspot.net/hubfs/700740/Reports/Newzoo_Free_2016_Global_Games_Market_Report.pdf.
[6] TNW Deals https://thenextweb.com/gaming/2016/03/24/engine-dominating-gaming-industry-right-now/#.tnw_C98G0uv2. Gaming, 2016.
[7] Unity – Company Facts https://unity3d.com/public-relations. 2017.
[8] Khoshelham, K.; Diaz-Vilarino, L.: *3D Modelling of Interior Spaces: Learning the Language of Indoor Architecture*. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-5, 321–326, 2014.
[9] Remondino, F.: *Heritage Recording and 3D Modeling with Photogrammetry and 3D Scanning*. Remote Sensing, vol. 3, no. 6, 1104–1138, 2011.
[10] Kaczmarek, A.L.: *Stereo vision with equal baseline multiple camera set (EBMCS) for obtaining depth maps of plants*. Computers and Electronics in Agriculture, vol. 135, pp. 23–37, 2017.
[11] Moulon, P.; Monasse, P.; Marlet, R. *et al.*: *OpenMVG. An Open Multiple View Geometry library*. https://github.com/openMVG/openMVG.

Mariusz Szwoch received his Ph.D. from the Faculty of Electronics, Telecommunications and Informatics of Gdańsk University of Technology in 2002. His research interests include image processing, pattern recognition and computer vision, interactive multimedia systems, computer graphics, and video games. He is the author of more than 60 peer reviewed publications.

Adam L. Kaczmarek received the Ph.D. degree in informatics in 2012 from Gdansk University of Technology, Gdansk, Poland. He also received the M. Sc., Eng. degrees in informatics from this university. Mr. Kaczmarek worked as an Intern at Intel Technology Poland from 2005 to 2007. Since 2007 he is working at Gdansk University of Technology. Currently, he is an Assistant Professor. His research interests include information retrieval, stereo vision and 3D vision.

Dariusz Bartoszewski received the M. Sc. degrees in economics from Gdansk University, Gdansk, Poland. He is a game developer with over 20 years of experience. Mr. Bartoszewski worked at Young Digital Poland and Aidem Media. Currently, he is a production manager at Forever Entertainment company.

Optimized material point method for real-time snow simulation

Paweł Morawski, Krzysztof Guzek

Lodz University of Technology, Łódź, Poland
krzysztof.guzek@p.lodz.pl

Abstract

In this paper, we present optimization of the material point method for enhancing the simulation of snow. Instead of a static Cartesian grid, we propose dynamic space subdivision using octree. As a result the grid nodes can be created only where necessary reducing the amount of calculations. Additionally our method adjusts the resolution of the single nodes depending on the material points count. This approach significantly accelerates the algorithm, providing the possibility to use it in real-time applications.

Index Terms

snow simulation, material point method, computer graphics

1 Introduction

Snow simulation is a challenging task in the interactive computer graphics due to the dynamic nature of this material and it's variety of forms. One of the latest researches in this field obtains convincing results of this natural phenomenon. However, current solutions are not yet satisfactory in case of real-time simulation.

Three methods have been examined in detail during this research: Smoothed Particle Hydrodynamics, Position Based Dynamics and Material Point Method (MPM). Based on the assumptions from the algorithm study, the MPM has been chosen, as the one with the highest potential for use in real-time snow simulation. The analysis shows that this hybrid technique, based on the Lagrangian particle and Euler grid implementation, has many advantages and returns very promising results. The most important problem though is the complexity of the calculations required by this method.

In connection with that fact, an improvement has been proposed based on the octree, which dynamically creates a grid and optimally adjusts the resolution of the nodes. The solution significantly accelerates the algorithm, providing the possibility to use it in real time applications. Some future improvements are also proposed.

2 Snow simulation methods

Snow accumulation as a part of snow geometric modeling has been investigated many times. One of the first proposed methods was presented in 1997 and based on metaballs [1], the multidimensional objects that look like organic models. The algorithm was not suitable for real time simulation and neglected snow dynamics.

The next step in the study of this area was the work of Paul Fearing [2]. The method used particles to simulate the accumulation of falling snow. In this approach the particles were fired from the ground upwards and then converted into a collection of smoothly merged three-dimensional surfaces. Rendering speed was low and the technique was suitable only for the initialization of the scene.

The method [2] was extended in the work [3] where the authors combined it with incomprehensible fluid flow techniques. The reduced version of the Navier-Stokes equation has been used to model the accumulation of snow. The result was a realistic scene of accumulation of snow by wind gusts.

Real time technique was described in the work [4]. The main idea was to use a two-dimensional matrix to store snow depth information in a particular area. The particle system was used to simulate falling snow and after collision with the ground was subjected to a triangulation process according to the height information obtained from the matrix.

Ohlsson and Seipel [5] adapted the depth-of-field method to calculate the amount of snow on each object on the scene and performed occlusion culling to render a snow scene quickly. However, this approach required a lot of time for preprocessing so the system was not quite efficient. Also nVidia company in technical report [6] described this method.

The combination of the Fearing model [2] with the falling snow animation was presented in [7]. The snow on the ground was represented by a grid of triangles while falling flakes were particles.

Foldes and Benes [8] proposed a solution for snow accumulation and partial melting simulations. Ambient occlusion was used to predict the shape and position of snow, and directional lighting for melting and sublimation. However, the technique focused on large areas visible from afar, such as mountains and was not suitable for showing snow dynamics.

Another method inspired by the methods [2], [3] and [7] described in the work [9] used the level set approach. The dual-level structure represent the dynamic surface of the snow and the static boundaries of the scene.

Snow animation are another issue that has been analyzed. Authors in the publication [10] used a height map (height field) for interactive animation of granular materials that could be deformed by contact with a rigid body.

Muraoka and Chiba in [11] added to the snow falling and accumulation also melting. Their formulas based on the microscopic, physical properties of snow and water. Snow humidity was used to check how snow will adhere to the object.

In the same year, method of snow accumulation was published by Festenberg and Gumhold [12]. The authors proposed a statistical model of snow deposition based on actual observations. The goal was to obtain a geometric form of the snow cover shape implemented with extended height maps.

Example of falling snow animation was presented in the work [13]. Method used a standard particle system to generate a rare collection of falling snowflakes and filled the gap between the particles using dynamic textures. The texture of the falling snow was defined by the dispersion in the plane of the image obtained from a linear perspective.

Saltvik parallel method [14] included the movement of snow in the air with the effect of wind, as well as the snow accumulation on objects and a ground in real time.

An interesting approach to the topic of snow simulation is presented in the article [15]. The authors used heat transfer between the various elements of the environment to create a realistic winter weather dependent scene. Voxels were used to represent the individual materials and to switch between states.

The phenomenon of snow is in computer games implemented in the form of predefined textures superimposed on objects used for accumulation on the surface and static models. Shader-based techniques are also known to provide satisfactory coverage of objects on the scene with snow. For rainfall visualization, use classical particle systems, billboarding, and other methods that do not reflect the physical model of snow.

Also, the analysis of the solutions available in the two leading-edge gaming engines has not added anything new to the subject. For Unity, the only available material is the general documentation for the embedded particle system [16]. On the other hand, Unreal Engine has prepared snow-related materials [17] but focuses on the visual side. In both cases, however, there are no advanced systems responsible for snow simulations, only methods mentioned earlier are used.

One of the more advanced solutions is implemented in the game "Batman: Arkham Origins" [18], which uses deformable snow rendering techniques. The algorithm allowed to generate deformable snow, which had low memory usage and computational cost. High-altitude mapping was used for this. In real time, an orthogonal clipping body is created that is close to the height of human cubes. All that is in this area affects the altitude map, which changes to the corresponding texture material if deformation occurs. This solution has allowed for satisfactory results, and dynamic allocation and assignment allowed for real-time operation.

Also in the case of "Company of Heroes 2" [19], there was a need for advanced winter scenery. As a result, a number of techniques were used in production: procedurally generated snow, automatic smoothing of snow edges and traces in the snow.

The snow accumulation used a properly crafted shader, which was based on a virtual snow throwing from the bottom to the top, based on the normal of the model walls. Smoothing of edges was solved by implementing a shadow map for snow accumulation. An additional layer of terrain created manually was used, which represented snow and allowed collapse of objects. Tracks in the snow were modeled using texturing techniques and a clipping plane at maximum snow height. Utilizing a whole set of techniques and the external software (Blizzard FX) has achieved satisfactory real-time results without physically simulating snow.

The Material Point Method was proposed first time in [20] and continued in [21]. The algorithm was extended in a subsequent publication of Sulsky and Schreyer [22] and named Material Point Method (MPM). The basic idea of this technique was a hybrid approach to calculating issues in the field of continuous mechanics. Using the advantages of the Euler and Lagrange methods, efficient algorithms have been developed in the context of materials stiffness, collisions, and topological changes.

The method has been extended in the work [23] for membrane simulations and in [24] for their interaction with liquids. Another proposed solution was used for granular material modeling [25], but the publication did not mention snow.

In 2013 a solution for snow simulation was proposed by Stomakhin [26] and has been extended [27] by the possibility of using other materials. Recent research [28] has focused on specific materials such as foam, sponges or viscous fluid.

Material Point Method based on a finite set of Lagrange material points (particles) moving along the Cartesian grid for the Euler equations. In the Euler formula, the superior equation is solved on a fixed grid and the material moves on it. In the Lagrange method, however, the mesh moves and distorts with the material.

The combination of the two descriptions allowed to avoid the disadvantages of the two solutions while preserving their advantages. The algorithm discretizes the continuous medium Ω to a finite collection of particles, which are tracked during the deformation process.

Simplified description of the algorithm consists of calculating the masses and forces of the particles acting on the given mesh node. In the next step, the velocity of the nodes is updated based on the equation of motion. After the increase in deformation and new particle velocities are calculated. The last step in the loop is to update the position of the particles based on the current speed.

The method [26] used for snow simulation was divided into an Euler part that operates on a Cartesian grid and a Lagrange's particles description of the snow. The result of calculations on the grid is applied to the particles, then the mesh is reset and it is possible to move on to the next step.

At the beginning of the simulation, it is necessary to create a simulation grid that defines the model boundaries, in the result the superior equation has boundary conditions and is solvable. Initial points of material and values such as density, initial velocity or mass are initialized. The first appropriate step in the simulation is to transfer the mass and particle velocity to the corresponding nodes of the grid. Previous values from nodes are not used because the mesh is reset at every time step.

3 Proposed optimization

The disadvantage of the Material Point Method is a high computational cost. One of the possible way to resolve this issue was presented in [29], another option worth considering was the transfer of calculations to the GPU as proposed in [30].

The solution proposed in this work does not use any concurrent nor GPU calculations technique. An improvement is based on the commonly known in computer graphics data structure – the octree.

Octree is a tree data structure used to index three-dimensional data [31]. Each node represents the volume formed from the cube. The method surrounds the space using cube. Then divides it into eight equal parts and then each of these parts into eight smaller parts. This procedure is repeated until the criterion of completing the algorithm is fulfilled. Each tree node has up to eight children, each corresponding to the connected octet of the node. Leaves (nodes without children) imply a uniformly represented volume that does not require further divisions. Storing points that do not have a related size require an algorithm stopping rule. Proposed solution uses the maximum depth of the method as well as the minimum number of points at which the leaf is created. The list of points is stored in each non-empty leaf.

In proposed solution the octree was designed in a classical way [31]. The management module only has access to the root, and dipping into the tree structure takes place through recursion until it encounters leaves. When building an octree, the two conditions described in [32] are used to balance the tree: the maximum tree level and the number of particles that should not exceed the leaf. As a result the created trees are not uniform and allow optimal space distribution in the context of material points.

The component had to be prepared to replace the Cartesian Euler grid so that the module interface was completed to fill all the steps required by the Material Point Method. To accomplish this, it was required to pass particles to the octree and distribute them to the tree nodes. Then perform the necessary calculations on the particles and leaves that simulate the nodes. The cycle ends with the return of the particles under the control of the particle system.

The method uses mentioned structure instead of a static Cartesian grid, as a result the grid nodes could be created dynamically only there where are necessary. Additionally the algorithm adjusts the resolution of the single nodes depending on the material points amount. The solution significantly accelerates the algorithm, providing the possibility to use it in real time applications.

The most important problem that caused most changes in the architecture of the solution was the dynamic initialization of octree and its deletion in each rendering frame. Particles vector was passed to the octree and was distributed to the next nodes in the tree. An effective solution was to redesign of the system behaviour so that instead of the entire data structures, the nodes were passed pointers rather than creating a deep copy of the collection.

Another worth mentioning problem was writing or using a ready-to-use solution for calculating the singular value decomposition of a third-order matrix that would be suitable for real-time use for large numbers of objects. In this case, the ready-made mathematical library Eigen is used, which has the appropriate module for performing such calculations, and can use external data structures. In addition, it is available under open source MPL2 license.

4 Results

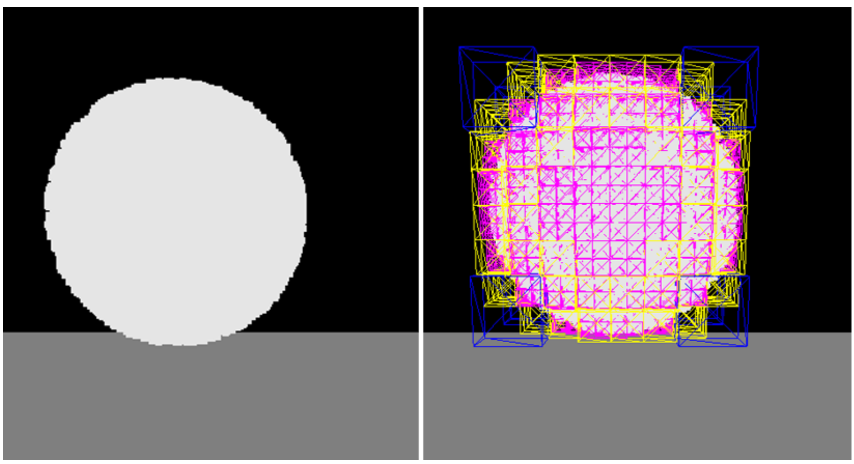


Fig. 1. Initial state of the simulation for 40k particles in the shape of the snowball (left) with octree visualization (right)

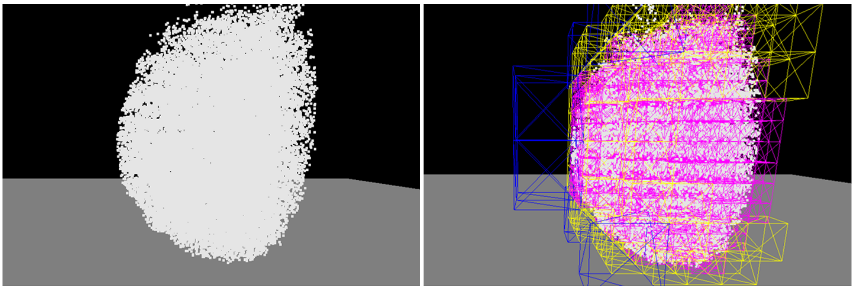


Fig. 2. Simulation of the snowball collision with the invisible wall (left) and octree visualization (right)

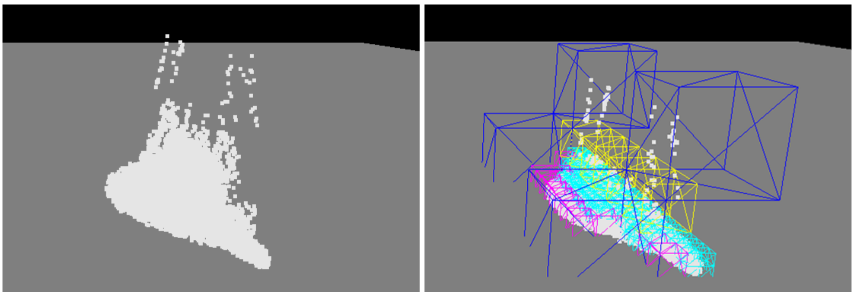


Fig. 3. Simulation of the snowball falling on the ground after collision with wall (left) and octree visualization (right)

All tests were ran on a quad-core Intel Core i5-4670K, 8GB of RAM, and an NVIDIA GeForce GTX 560Ti graphics card with 1GB of memory in Windows 10. Simulator was implemented in C++ 11 with OpenGL and freeglut. Microsoft Visual Studio 2015 was used as an integrated development environment.

In order to verify the proposed approach two different test simulations were performed. The first simulation is the snowball's movement combined with the collision. Snow moves to the right until it contacts the wall and falls on the ground. Three main stages of this simulation performed for the scene with 40,000 particles are presented in Figures 1-3 with enabled octree visualization.

Next simulated scene is snow falling by the force of gravity combined with a rigid body collision in the form of a sphere. Snow moves down in accordance with earthly gravity and during this movement a collision with the body occurs. In the scene, 20000 particles were used, and the result of the simulation is presented in Figures 4-6. The effects are shown in three steps corresponding to the initial simulation, the moment of collision and the final result. Also here, octree visualization is included.

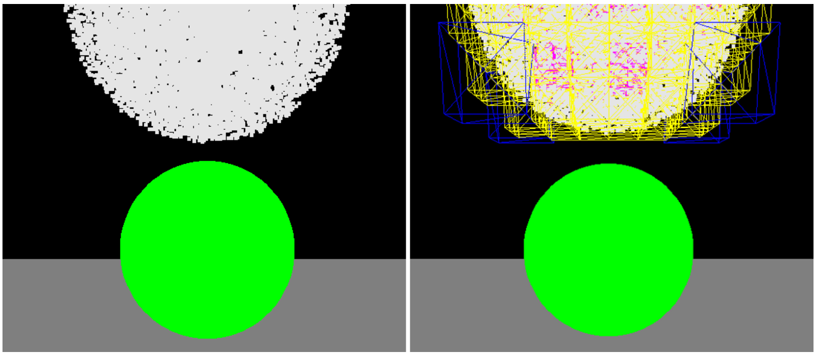


Fig. 4. Initial state of the simulation for 20k particles falling on a rigid body in the form of a sphere (left) and octree visualization (right)

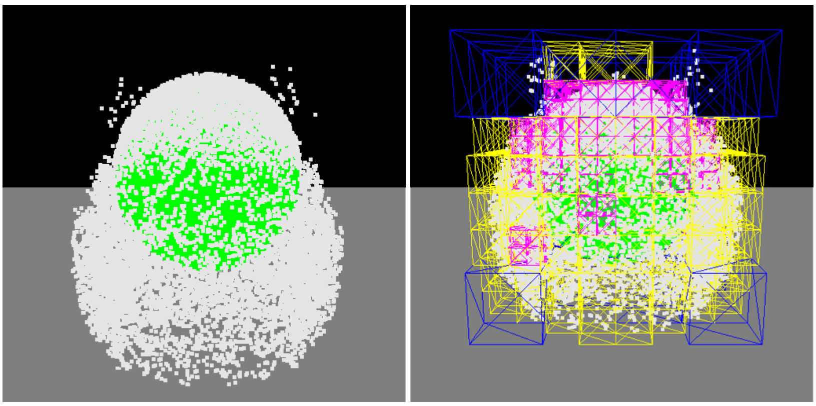


Fig. 5. The moment of snow collision with the sphere (left) and octree visualization (right)

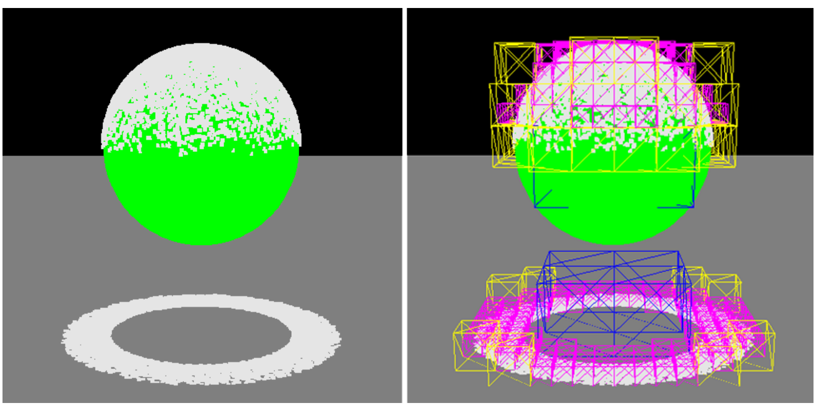


Fig. 6. The final result of the simulation: visible snow accumulation on the sphere and on the plane below forming a ring shape (left) with octree visualization (right)

The bar chart shown in Figure 7 visualizes the minimum, arithmetic mean and maximum frame rate (FPS) received in the application with different numbers of simulated particles. Values were collected from four separate runs for each number of particles visible in the graph. Each single launch lasted about 30 seconds during which the snowball moved along the scene until it collided with the wall and the snow fell to the ground.

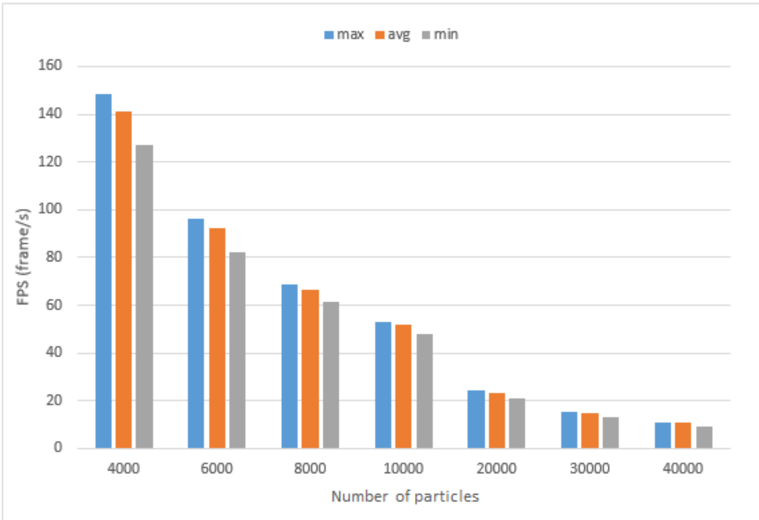


Fig. 7. Frame rate versus the number of simulated particles for Optimized Material Point Method

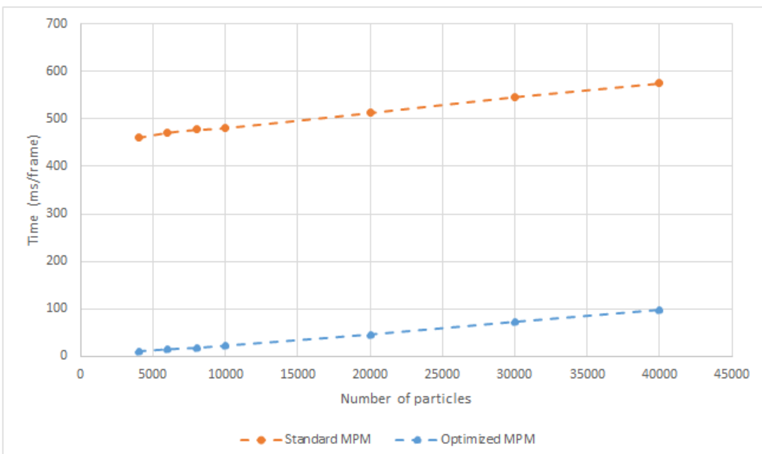


Fig. 8. The average time spent generating a single frame at the specified number of particles used in the simulation along with the trend line for Standard Material Point Method and Optimized Material Point Method

The graph in Figure 8 shows the average time in milliseconds that is needed to generate a single simulation frame. The results were collected by the same method as described above in the previous graph.

The impact of grid nodes on simulation performance at constant number of particles set at 15000 was also tested. However, in contrast to particles, nodes are dynamically generated by the octree. To collect the data shown in Figure 9, it was necessary to manipulate the maximum number of particles in a single node. The following values were set: 3000, 700, 500, 100, 10, 5, 1. In this case, four simulations were also run for each variant and the data collected were averaged both for FPS and nodes amount.

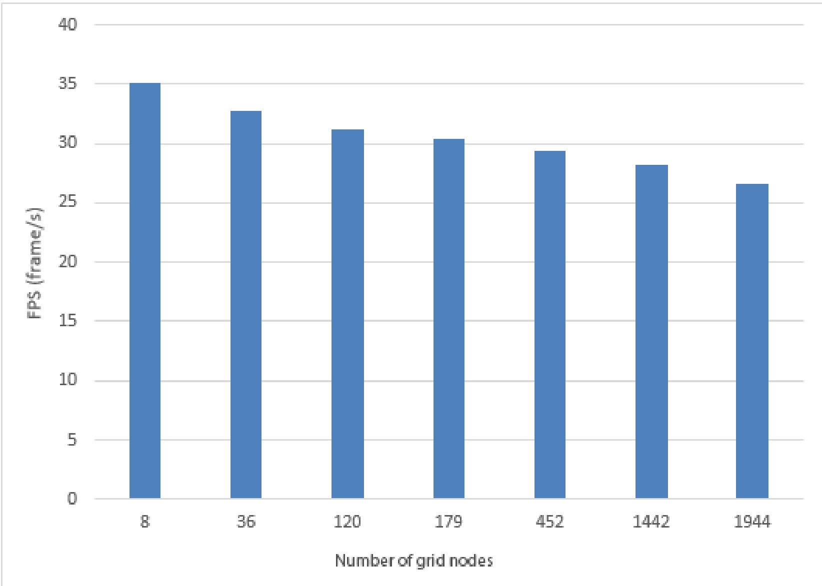


Fig. 9. Frame rate versus the number of grid nodes with a constant amount (15k) of particles for Optimized Material Point Method

It has also been verified how the number of grid nodes at a constant number of particles affects the quality of the simulation. Results are shown in the Figure 10.

5 Discussion

The performance of the proposed solution is dependent on the number of particles and grid nodes used for the snow simulation. Higher resolution of the grid provides a better representation of the surface of the simulated model. In contrary, increasing the number of particles allows for mapping larger objects. However, this results a significantly increasing number of operations required at each frame, since more particles and nodes require the necessary calculations. Therefore, the performance of simulation is decreasing.

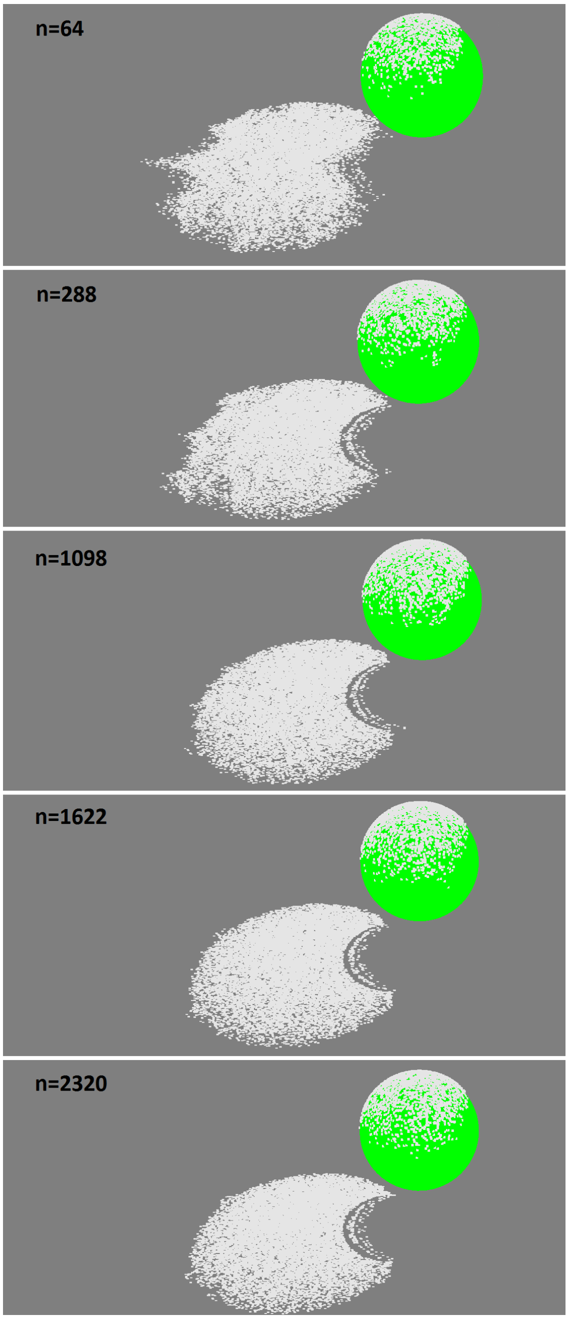


Fig. 10. Comparison of snow accumulation simulation at the initial number of nodes (n) equal 64, 288, 1098, 1622 and 2320

The diagram shown in Figure 7 allows to assess the relationship between the frame rate per second and the number of simulated particles. The optimal results for real time simulations are obtained to the limit of about 15,000 particles when the simulation still achieves a minimum of 30 frames per second. Further results are already showing the visual effects of too low FPS.

Based on the Figure 8 one can compare time results obtained for Standard Material Point Method and Optimized Material Point Method for the growing number of particles used in the experiment. According to the graph, for both metohods the average time increases linearly with the growing number of particles, however optimized approach takes almost of 500 ms per frame less than standard method. This linear relationship indicates that each single particle simulation takes approximately the same amount of processing time in a separate frame, regardless of the number of particles.

Still, calculating the SVD of the matrix is the process that consumes the most CPU time and resources in the performance context. The possible solution is to move these calculations onto the graphics card.

The results show that the impact of amount of grid nodes on simulation is small, which is an advantage of using the octree instead of Cartesian static grid. The best results in performance to quality were obtained at 300 to 500 nodes. The dramatic reduction in grid resolution, as shown in Figure 10, significantly influences snow accumulation. After exceeding the value of 1200 nodes, it is hard to capture any visual differences as shown in Figure 10. The results helped to optimize the algorithm and achieve better results without sacrificing performance.

6 CONCLUSIONS AND FURTHER WORKS

Material Point Method and its derivatives has a lot of benefits. The major one is realism. This method allows to simulate different types of snow from dry, dense to wet, taking into account physical properties and making the result compatible with physical models.

The disadvantage of this solution is performance. The calculations for very high number of particles often leads to large amounts of allocated memory and processor time. Unfortunately, this reduces the possibility of using this technique in real-time applications such as computer games.

Even the improvements proposed in this work do not fully address this problem. They allows, however, to use this method in real time without having to render the scene in advance and seem a promising introduction to further research. The possible solution to this problem is to move some or all of the calculations to a graphics card, which should speed up the simulation and allow to simulate a much larger number of particles.

Especially the singular value decomposition of three dimensional matrix should be transfered to GPU, because in implementation of proposed solution this computational step required most of processor time. Also, further investigation and optimization of dynamic creation, management and removal of grid cells as well as particles could reduce render time for a single frame.

Another aspect that should be explored further is the rendering of snow and its effective implementation to the algorithm presented here. Thus realistic physical behavior will be accompanied by the real visual impressions.

References

[1] Nishita, T.; Iwasaki, H.; Dobashi, Y.; Nakamae, E.: *A Modeling and Rendering Method for Snow by Using Metaballs*, Computer Graphics Forum, vol. 16, no. 3, pp. 357-364, 1997.

[2] Fearing P.: *Computer Modelling of Fallen Snow*, 27th Annual Conference on Computer Graphics and Interactive Techniques, New York, 2000.

[3] Feldman, B. E.; O'Brien, J.F.: *Modeling the accumulation of wind-driven snow*. SIGGRAPH '02: ACM SIGGRAPH 2002 conference abstracts and applications, New York, 2002.

[4] Haglund, H.; Andersson, M.; Hast, A.: *Snow accumulation in real-time*, SIGRAD, Gävle, 2002.

[5] Ohlsson P.; Seipel S.: *Real-time rendering of accumulated snow*, Proceedings of SIGRAD, Gävle, 2004.

[6] Dudash, B.: *Snow Accumulation*, NVIDIA Corporation, Santa Clara, 2005.

[7] Moeslund, T. B.; Madsen, C. B.; Aagaard, M.; Lerche, D.: *Modeling falling and accumulating snow*. Second International Conference on Vision, Video and Graphics, Aalborg, 2005.

[8] Foldes, D.; Bedrich, B.: *Occlusion-based snow accumulation simulation.*, 4th Workshop in Virtual Reality Interactions and Physical Simulation VRIPHYS, 2007.

[9] Hinks, T.; Museth, K.: *Wind-Driven Snow Buildup Using a Level Set Approach*, Eurographics Ireland Workshop Series, vol. 9, pp. 19-26, 2009.

[10] Sumner, R.W.; O'Brien, J.F.; Hodgins, J.K.: *Animating sand, mud, and snow*. Computer Graphics Forum, vol. 18, no. 1, pp. 17-26, 1999.

[11] Muraoka, K.; Chiba, N.: *Visual simulation of snowfall, snow cover and snowmelt*. Parallel and Distributed Systems: Workshops, Seventh International Conference on 2000, Morioka, 2000.

[12] v. Festenberg, N.; Gumhold, S.: *A Geometric Algorithm for Snow Distribution in Virtual Scenes*. NPH'09 Proceedings of the Fifth Eurographics conference on Natural Phenomena, Munich, 2009.

[13] Langer, M. S.; Zhang, L.; Klein, A.; Bhatia, A.; Pereira, J; Rekhi, D.: *A spectral-particle hybrid method for rendering falling snow*. Eurographics Symposium on Rendering, 2004.

[14] Saltvik, I.; Elster, A.C.; Nagel, H. R.: *Parallel methods for real-time visualization of snow*. Applied Parallel Computing. State of the Art in Scientific Computing., Berlin, Springer-Verlag, pp. 218-227, 2008.

[15] Maréchal, N.; Guérin, E.; Galin, E.; Mérillou, S.; Mérillou, N.: *Heat transfer simulation for modeling realistic winter sceneries*. Computer Graphics Forum 29, vol. 2, pp. 449-458, 2010.

[16] Unity Technologies: *Unity – Manual: Particle System*. https://docs.unity3d.com/Manual/class-ParticleSystem.html, August 25th 2017.

[17] Epic Games Inc.: *Snow Examples | Unreal Engine*. https://docs.unrealengine.com/latest/INT/Resources/Showcases/Effects/SnowExamples/, August 25th 2017.

[18] Barre-Brisebois, C.: *GDC Vault – Deformable Snow Rendering in Batman: Arkham Origins*. http://www.gdcvault.com/play/1020177/Deformable-Snow-Rendering-in-Batman, August 25th 2017.

[19] Barrero, D.: *Company of Heroes 2 (COH2) Rendering Technology: The cold facts of recreating the hardest winter conditions of World War 2*. Korea Game Conference, 2013.

[20] Sulsky, D.: Chen, Z.; Schreyer, H.L.: *A particle method for history-dependent materials*. Computer Methods in Applied Mechanics and Engineering, vol. 118, no. 1-2, pp. 179-196, 1994.

[21] Sulsky, D.; Zhou, S.J.; Schreyer, H.L.: *Application of a Particle-in-cell method to solid mechanics*. Computer Physics Communications, vol. 87, no. 1-2, pp. 236-252, 1995.

[22] Sulsky, D.; Schreyer, H.L.: *Axisymmetric Form of the Material Point Method with Applications to Upsetting and Taylor Impact Problems*. Computer Methods in Applied Mechanics and Engineering, vol. 139, no. 1-4, pp. 409-429, 1996.

[23] York, A.R.; Sulsky, D.; Schreyer, H.L.: *The material point method for simulation of thin membranes*, International Journal for Numerical Methods in Engineering, vol. 44, no. 10, pp. 1429-1456, 1999.

[24] York, A.R.; Sulsky, D.; Schreyer, H.L.: *Fluid-membrane interaction based on the material point method*, International Journal for Numerical Methods in Engineering, vol. 48, no. 6, pp. 901-924, 2000.

[25] Bardenhagen, S.G.; Brackbill, J.U.; Sulsky, D.: *The material-point method for granular materials*. Computer methods in applied mechanics and engineering, vol. 187, no. 3, pp. 529-541, 2000.

[26] Stomakhin, A.; Schroeder, C.; Chai, L.; Teran, J.; Selle, A.: *A Material Point Method for Snow Simulation*. ACM Transactions on Graphics, vol. 32, no. 4, pp. 1-10, 2013.

[27] Stomakhin, A.; Schroeder, C.; Jiang, C.; Chai, L.; Teran, J.; Selle, A.: *Augmented mpm for phase-change and varied materials*. ACM Transactions on Graphics, vol. 33, no. 4, pp. 138, 2014.

[28] Ram, D.; Gast, T.; Jiang, C.; Schroeder, C.; Stomakhin, A.; Teran, J.; Kavehpour, P.: *A material point method for viscoelastic fluids, foams and sponges*. Proceedings of the 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 157-163, 2015.

[29] Shin, W.; Miller, G. R.; Arduino, P.; Mackenzie-Helnwein P.: *Dynamic meshing for material point method computations*. International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering, vol. 4, no. 8, pp. 84-92, 2010.
[30] Dong, Y.; Wang, D.; Randolph, M. F.: *A GPU parallel computing strategy for the material point method*. Computers and Geotechnics, vol. 66, pp. 31-38, 2015.
[31] Elseberg, J., Borrmann, D.; Nuchter, A.: *One billion points in the cloud – An octree for efficient processing of 3D laser scans*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 76, pp. 76-88, 2013.
[32] Gomes, A.; Voiculescu, I.; Jorge, J.; Wyvill, B.; Galbraith, C.: *Implicit Curves and Surfaces: Mathematics, Data Structures and Algorithms*. London, Springer-Verlag, 2009.

Paweł Morawski Paweł Morawski received the BSc (2015) and MSc (2017) in computer science from Lodz University of Technology, Poland. He joined the Accenture Company, first as an intern in 2013 and later on as a full time employee. In 2016, he joined the Cybercom Poland Company, where he is presently a Software Engineer.

Krzysztof Guzek Krzysztof Guzek received his PhD (2014) in computer science from Lodz University of Technology, where he currently works as a lecturer and researcher in the unit of Computer Graphics and Multimedia. His research focuses on three-dimensional computer animation, photo-realistic image synthesis and global illumination algorithms.

The **"Computer Game Innovations"** series is an international forum designed to enable the exchange of knowledge and expertise in the field of video game development. Comprising both academic research and industrial needs, the series aims at advancing innovative industry-academia collaboration. The monograph provides a unique set of articles presenting original research conducted in the leading academic centres which specialise in video games education. The goal of the publication is, among others, to enhance networking opportunities for industry and university representatives seeking to form R&D partnerships. This publication covers the key focus areas specified in the GAMEINN sectoral programme supported by the National Centre for Research and Development.

As the host of the **Game Creation Challenge** competition (formerly "Konkurs Zespołowego Tworzenia Gier Komputerowych") and the **Conference on Game Innovations**, the Faculty of Technical Physics, Information Technology and Applied Mathematics of Lodz University of Technology has the pleasure of leading this well-established process of knowledge and experience exchange. For one day, the city of Łódź brings together distinguished specialists and researchers from all over the world. The Game Creation Challenge is a renowned proving ground for young talented game developers, while the Conference provides a forum for the exchange of expertise between academic and professional experts. These initiatives are designed to suit the needs of the rapidly expanding game development industry, which has grown to become a leading branch of the creative industry worldwide, especially in Poland.

Politechnika Łódzka

INSTYTUT INFORMATYKI POLITECHNIKA ŁÓDZKA
ul. Wólczańska 215, 90-924 Łódź

Wydział
Fizyki Technicznej
Informatyki
i Matematyki Stosowanej

ISBN 978-83-7283-868-1