KINACT: A SALIENCY-BASED SOCIAL GAME

Matei Mancas 1, Radhwan Ben Madhkour 1, Dominique De Beul 1, Julien Leroy 1, Nicolas Riche 1, Yves P. Rybarczyk 2, François Zajéga 1

1 Laboratoire de Théorie des Circuits et Traitement du Signal (TCTS), Université de Mons (UMONS), Belgique
2 Departamento de Engenharia Electrotécnica, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, Portugal

ABSTRACT

In this paper we discuss the design of an intelligent game system capable of selecting the players who exhibit the most outstanding behavior from groups of people playing on a network. Players can be all in the same place (maximum 3) or in several locations if connected to the web. In that way, the total amount of players can be very large. Tests were made with 3 players in a single location and with 2 different locations of 2 players each. The system uses both static and dynamic features extracted from the upper part of the players’ bodies such as symmetry, contraction index, motion index or height. Those features are extracted using the RGB-D Kinect sensor and their relative contrast and time evolution enable an adaptive selection of the most salient or different behavior without any complex rules. First users’ feedback and eye tracking tests are shown and applications to social interactions are presented.

KEYWORDS

saliency, computational attention, game, kinect, bottom-up, top-down, behavioral features, eye tracking, social interaction

1. INTRODUCTION

Efficient and low-cost devices as the Kinect sensor opened new highways in gaming community. People detection and skeleton extraction in real life light conditions becomes a reality and more and more complex interactions are possible. Despite those new possibilities, still the interactions are achieved with pre-learnt and pre-programmed gestures. This means that if game scenarios are different from what is expected, the system cannot cope with the novelty.

An impressive human ability is to be able to attend interesting events or interesting behavior even if the situation is completely novel. In this paper, we implement and demonstrate a game which is able to select the active player in an attentive way. The system only uses players’ behavior during their observation and a short memory of what they previously did to select in any situation the active player. The game is a network implementation using results from [9].

Beyond the ability to choose the active user, the system will be used to study the social organization of the player team, the influence of the players’ localization which can be different between people playing together or through the network [13].

The characteristics we use are mainly related to 3D motion but also social/emotional features as those developed by Camurri et al. [3]. Two of the four features used namely the motion index (also called quantity of motion) and the contraction index come from those developments. The four features used here describe well the kind of gestures that humans would naturally perform in a game/social context like shaking hands, move, sit down...

This paper is organized as follows: section 2 globally discusses the game idea and its play modes. Section 3 details the features extraction while section 4 presents both bottom-up and top-down mechanisms. Section 5 deals with the implementation of our game and especially the network issues. Section 6 provides some results to tests which intend to capture players’ interest for this concept and the accuracy of the attention system. Section 7 provides a conclusion and further developments of this game which intends to be a platform towards social behavior analysis.

2. GAME DESCRIPTION

2.1. Idea and Interface Implementation

We based our game on “World of Goo” [2], a puzzle game developed by 2D Boy [1]. The purpose proposed to players in World of Goo is to build a structure made out of balls interconnected by semi-rigid wires. This structure is used to touch a goal located somewhere in the scene.

We implemented a similar game by using the physical interaction Box2D library [4] and QT [10]. Figure 1 shows a screenshot of the game where two players almost touched the goal which is here a top-screen yellow ball.

Figure 1: Two players made a structure of white balls linked by elastic black wires and the purpose is to touch the top yellow ball.
the structure. The game finishes when there are no more balls left or when the goal is reached.

2.2. Use Players’ Body as a Controller

Two games modes were implemented. The first one is a simple sequential mode where each player has 10 seconds to put his own ball, and then the next one becomes active and so on. This is a classical implementation which uses no intelligence in selecting the active user.

The second game mode uses automatic saliency based on players’ behavior to attend to a particular player and make him active. There are two distinct steps in each game turn. First, during 5 seconds, the players have to differentiate themselves from the others by adopting unfamiliar or strange poses (to attract the attention of the system). In a second step, as for the sequential mode, the active player has 10 seconds to place his ball on the structure.

While in the first mode at least two players are needed, in the second one at least 3 are needed to find the player who is different from at least two others. Each player is represented graphically as a dashed circle. He controls the position of the cursor by moving in front of the Kinect sensor: the X and Z coordinates of the skeleton’s barycenter of each player is mapped on the X and Y position of the cursor. In the second mode, behavioral features are extracted from the upper part of the skeleton of each player.

3. FEATURES EXTRACTION

The first step for an interactive machine is to extract features from the observed people. For that purpose, we use the Kinect sensor for its ability to extract smooth depth maps in complex illumination conditions. Libraries as OpenNI [8] are available to detect human silhouettes and extract anatomical features from skeletons tracking (Figure 2).

Figure 2: Real-time skeleton extraction from players using OpenNI library.

Four features are extracted from the upper body part only as the legs are much less stable in our implementation.

One of the four features is dynamic, namely the motion index. It is computed as the mean variation of the same skeleton points between two frames in 3D (on X, Y and Z). The barycenter point variation is extracted from the others (Equation 1) in order to keep only the body relative motion which will describe an excitement degree or movement transition of the body without any assumption on the whole body speed.

\[
D_{mk} = \left\{ \left( \sum_{sk} |k_b - k_{sh}| \right)_{t} - \left( \sum_{sk} |k_b - k_{sh}| \right)_{t-1} \right\} \\
\text{where } k = x, y, z ; sk = \text{skeleton points} \\
\text{and } b = \text{Barycenter}
\]

\[
MI = \sqrt{D_{mx}^2 + D_{my}^2 + D_{mz}^2}
\]

A second feature extracted from the upper body part is a static feature, namely the asymmetry index. This feature is only computed on the X axis by differencing the distances between the barycenter point and the right shoulder, elbow and hand points with the left ones (Equation 2). This index provides information about the symmetry of the upper body.

\[
AI = \frac{\sum_{sk} |X_b - X_{sk\text{right}}| - \sum_{sk} |X_b - X_{sk\text{left}}|}{n_{sk}}
\]

where \( n_{sk} \) = number of skeleton points

The third extracted feature is the contraction index. This index is the ratio between the maximal distance between skeleton points on X axis and the maximal distance on the Y axis (Equation 3). This index tells us if the person is more or less contracted.

\[
CI = \frac{\text{max}(X) - \text{min}(X)}{\text{max}(Y) - \text{min}(Y)}
\]

The fourth and final feature is the player height. That one is simply computed by measuring the player barycenter Y coordinate.

After normalization, those four features provide a quite complete description about the level of excitement, and the upper body configuration of each player.

4. COMPUTATIONAL ATTENTION: WHO IS DIFFERENT?

The aim of computational attention is to provide algorithms to automatically predict human attention. The term of attention refers to the whole attentional process that allows one to focus on some stimuli at the expense of others. Human attention is mainly divided into two main influences: a bottom-up and a top-down one. Bottom-up attention uses signal characteristics to find the most salient or outstanding objects. Top-down attention uses a priori
knowledge about the scene or task-oriented knowledge in order to modify (inhibit or enhance) the bottom-up saliency. The relationship and the relative importance between bottom-up and top-down attention is complex and it can vary depending on the situation [8]. Much research has already addressed the computational visual attention, but these works were devoted almost exclusively to 2D image analysis. Up to now little has been done on 3D data [11]. The arrival on the market of low cost 3D cameras opens up new prospects for developing algorithms for visual attention, which should make the move towards more realistic ambient intelligence systems.

4.1. Bottom-up Approach

As stated in [7] a feature does not attract attention by itself: bright and dark, locally contrasted areas or not, red or blue can equally attract human attention depending on their context. In the same way, motion can be as interesting as the lack of motion depending on the context. The main cue, which involves bottom-up attention, is the contrast and rarity of a feature in a given context.

The approach here follows the one in [9] but it has an optimal implementation in C++ instead of several pieces of block-processing software and it is also adapted to games and a network use. In our case, as the group of players can be small, the rarity computation is not relevant; therefore we only use the global contrast. Thus, the first step in this section is to calculate the $i^{th}$ feature ($f_{i,k}$) a contrast between the different users $k$.

$$C_{i,k} = \sum_{j=1}^{N} \frac{|f_{i,k} - f_{i,j}|}{N-1}$$

where $N$ is the number of users. Once all the contrasts for a given feature $C_{i,k}$ between each user and the others have been computed, they are ordered in ascending order $C_{i,k,o}$ with $o = [1 : N]$ from the maximum ($o = 1$) to the minimum ($o = N$). The difference between the two highest values is compared to a threshold $T$ which decides if the contrast is large enough to be taken into account as in Equation 5.

$$\begin{cases} \alpha = 0 & \text{if } |C_{i,k,1} - C_{i,k,2}| < T \\ \alpha > 0 & \text{if } |C_{i,k,1} - C_{i,k,2}| \geq T \end{cases}$$

Only the features being the largest and passing this threshold $T$ are merged with different weights (Equation 6).

$$C_k = \sum_{i=1}^{H} \frac{|C_{i,k} \ast W_i \ast \alpha|}{H}$$

Where $H$ is the number of features and $\alpha$ is given in Equation 5. The way the values of the weights $W_i$ is set is described in the next section. This contrast $C_k$ represents the bottom-up saliency for each user $k$. Saliency will be higher for the people exhibiting the most contrasted features within a given frame. The process of bottom-up attention is summarized on Figure 4 on a three-player scenario example. Each of the three players has its four features computed (in red for the asymmetry index, yellow for the contraction index, violet for the motion index and green for the height). The contrast computation and thresholded (Equations 4 and 5) is displayed in the second column. Finally the contrasted features combination (Equation 6) is explained in the third and fourth columns.

4.2. Top-down Influence

The weights from Equation 6 can be adapted via the top-down component of attention. Those weights are initially set to be the same for all the 4 features which are used here. Then, the number of times a feature is contrasted enough for a given user ($\alpha > 0$), a counter is increased. The feature weight will be inversely proportional to its counter: if a feature is often contrasted, its weight will be lower and lower, while a feature which is rarely contrasted enough will see its weight increased. This mechanism ensures a higher weight to novel behavior while too repetitive behavior will be penalized.

As an example, someone who will sit down for the first time (different height feature compared to the others), the height will have the maximum weight. If this person thinks that a different height is enough to attract the system attention, he will try again, but the more he tries again, the more the height feature weight will decrease as this behavior is no longer surprising. This top-down approach allows the system to learn how much a feature is novel and provides higher weights to the most novel ones. While the bottom-up approach will analyze in each frame which person behaves in a different way compared to the others, the top-down approach will also look for people who behave in a novel (or less repetitive) way from the beginning of the observation by taking into account a time memory.

5. IMPLEMENTATION: A DELOCALIZED GAME

While network implementations of games are interesting per se, in the case of an attentive system, the interest is even higher as the more people are involved, the more the attentive selection approach is relevant. We thus implemented our game in a network framework by using a client/server architecture.
5.1. A Client/Server Architecture

The range of each Kinect is quite narrow. After 4.5 meters, the depth map is less consistent and the skeleton extraction out of this depth map becomes very unstable. At 4.5 meters from the Kinect, the field of view of the camera is about 6 meters large. The number of people who can be involved in a game using only one Kinect is restricted to 3 or 4 maximum. Due to the attention system, the players have to move a lot and perform gestures to gain the focus. This is an additional restriction on the number of users to ensure a good comfort during the game. If players are too close, they might hit themselves, and the skeleton tracking, in case of people touching their neighbors, are much less reliable. That is why we tested a maximum of 3 players per Kinect in this game configuration, which is also the minimum number of users required by an attentional system.

Therefore, we prepared a software architecture that allows the usage of several Kinect cameras concurrently to cover a larger space [9]. But this technique is more complex to set up and needs several sensors, a calibration process and specific software. That is why, another solution was adopted here: the game was prepared to be able to be used in the same time in different locations over the Internet. In this way, the minimum number of 3 players has to be reached in the whole game which means that in one location, there might be only two players and in a second location just one player is enough. We made tests with a configuration of two different locations and 2 players per location. This setup is shown in Figure 5. The two locations are here two different rooms in the same building, but the data pass through Internet and not on the local network. Preliminary tests were also made between the universities of the city of Mons in Belgium and the city of Pilsen in Czech Republic distant about 800 kilometers.

![Figure 5: Test with 4 players in two different locations (left and right image). While the visual feedback is visible for the first location in the left image, the middle image shows the visual feedback from the second location. Both structures are exactly the same.](image)

In order to implement this network approach, we had to choose between a peer-to-peer and a centralized architecture. The client-server architecture has been chosen for two main reasons.

The first one is because the game elements are managed by a physical engine: Box2D. Running this library on each client always led to inconsistencies. The longer applications were running the more difference appeared in connected clients because even a very small asynchrony leads to differences in the physical engine. In this case, placing the calculation of the graphical elements on a central server and broadcasting the elements to render to each client was the most effective solution.

The second reason is that each element involved in the game has a unique identifier. The management of these identifiers is better achieved by an entity controlling all the connected applications.

In the final version of the game (Figure 6), each client preprocesses the information. The retrieval of the skeletons of the players via OpenNI and the extraction of the behavioral features are executed on each client. The information is then packed in a message and sent to the server.

The server collects these messages and computes the final attention by comparing the behaviors from the different clients. After skeleton extraction on the client-side, each player has a local identifier. To clearly identify each player system-wide, the server distributes global unique identifier to each player. Each client manages the mapping between local and global ID’s. In the same time, the server processes a new state of the physical world and manages the unique IDs of the players from all the clients. Eventually, it prepares messages containing the graphical elements and the users to draw on the screen customized for each client and send them back to the clients.

When clients receive the “frame” message, they control that the frame is newer then the one previously received and adapts the graphical display with it.

![Figure 6: The client-server architecture of the game. The clients make the signal acquisition and compute the behavioral features of each player. They send the players’ IDs, position and features to the server. The server computes the attention and send back the active user and the changes into the display computed by the physical engine. Finally, the clients display the updated visual feedback.](image)

5.2. Mixed TCP/UDP Communication

Gaming over the Internet implies fast data transmission between all the components of the application. We wanted to guarantee 30 frames per seconds on the client-side so we had to prepare an optimized data transfer using the right protocol, TCP or UDP.

After testing both separately, we built a system using both for what they are good for. A TCP connection is used to start the client-server communication. No data is sent through this connection. It is used only to ensure the presence of both parties. If the connection breaks, the UDP connection will stop instantaneously.

A UDP connection is started directly after the validation of the TCP one. Its speed ensures a good frame rate and responsiveness. This solution is very effective and robust. Not sending any data trough TCP reduces to minimum potential loss of packages [12].

5.3. Players Behavior

The use of this network architecture has a lot of advantages by increasing the number of players without decreasing the game comfort. Nevertheless, as the attention algorithms is based on having a
different behavior compared to the other players, any player should be aware about the others. This is obvious when playing in the same location, but much less when the game is delocalized. The network architecture allowed us to send in real-time feature packets, but sending videos is much more complex and this was not in our focus. Therefore we decided to use icons to show the feature which is the most representative of each player. Figure 7 shows the icons which are displayed for each user depending on his maximum feature. Those icons are explicit enough to be immediately understood by the players. If no feature is above a threshold, the gray static icon is displayed.

Moreover, after the game is finished, it is possible to save in a XML file data about the features used, the number of balls, the joints between balls, the number of broken joints, the time needed to finish the game. This data showed that there is a faster progression in the learning of the game in the sequential mode than the attentive one which needs more time to be understood and even more to find efficient tactics. Indeed, this mode needs to build a social structure in the team with a leader and specialized players, while the sequential mode needs no specific involvement as all the players have the same role and the only solution to play is to wait your turn. In the sequential mode, people learn the game very fast and in only 3 trials, a team of 3 people went from 460 seconds and 39 balls needed to finish the game to 250 seconds and 23 balls. For the attentive mode, people went from 627 seconds and 36 balls to 501 seconds and 31 balls after a dozen of trials.

6. FIRST ANALYSIS

After the first tests, some analyses were carried out in the game. In a first step we asked the players to fill a questionnaire and we also used data which was available as output of the game. In a second step we used eye tracking to validate the attentive player selection system.

6.1. Users’ Feedback and Behavior Analysis

In order to have a first idea about the players’ feelings we set up a questionnaire asking them to score their game experience in terms of 1/Enjoyment, 2/ Isolation compared to the others, 3/ Comfort, 4/ Cooperation, 5/Leadership, 6/ Communication for both versions of the game: the classical sequential one, and the attentive one.

The level of comfort was higher in the sequential mode and the enjoyment comparable (probably also due to the fact that the game was more stable in the sequential mode than the attentive one during those tests). All the other factors were higher for the attentive approach. The players felt the attentive scheme as leading to much higher cooperation, leadership, communication and less isolation which is due to the fact that to attract the attention of the machine, the player has to take care not only about himself but also about the others.

Figure 7: Icons representing the dominant feature of the players. If there is no specific feature above a threshold, the static gray icon is displayed. Then, the feature which has the maximum intensity for each user is displayed.

6.2. Attention Selection: Eye-tracking Study

To validate our attention-based selection system, we can test what would catch the attention of a human put in the same conditions as the proposed game system. For this purpose, we simulated a typical three-player scenario located in the same place. The RGB stream as viewed by the system was recorded and displayed on a screen where eye tracking [6] was performed using a commercial FaceLab system. The 10 users who took part to the test were people from 25 to 35 years old, 3 females and 7 males. They were not aware about the game principle and they were just asked to watch in a natural way a video where 3 players were playing (black background to avoid stimuli coming from the background). No particular task was given to the users, thus their attention was mainly driven by bottom-up features even if top-down influence was also present as one of the three players in the video gazes to another player which also push the eye-tracked users to look in the same direction.

Within the bottom-up features, the static ones like color were used only at the very beginning of the observation, so during most of the video observation, users used bottom-up motion and position features. The results of this test (Figure 8) is surprisingly well correlated to contrasted people behavior, which is also the mechanism used by the proposed system. Of course this test setup uses one contrasted feature at each time and one of the three players is clearly contrasted, while in reality when people play, who is the most contrasted is sometimes not obvious even for humans.

The inter-personal distance is a very interesting feature which has clearly an effect on human attention (Figure 9) that we took into account in [9] but within the game with no network the feature was not easy to apply. The three players do not have enough lateral space to really change the inter-personal distances.

7. DISCUSSION AND CONCLUSION

In this paper we presented an original way of selecting an active user in a network game. Visual attention is implemented into the system which becomes able to choose the active player not by using pre-programmed paradigms but by adaptively finding the most outstanding player in any situation.

This work shows an important improvement potential. The first task will be to make more tests on user perception on the new game which is much more stable and simpler to understand than for the initial tests with the players’ questionnaire. A better implementation and deeper analysis of the data saved into the XML file concerning the ball structure construction is also needed. From
the XML file it is for example possible to extract the characteristics of the ball structure shape (Figure 10). A very large basis and a displacement of the balls during its construction may signify that the structure collapsed during the game, so the construction was not very efficient. Colors can be assigned to each user to see chat tactics were used to get information about the tactics used for example.

Within the new network development, it would be interesting to integrate the inter-personal distance feature in the virtual space of the game and monitor people positioning when they are located in the same place or in different locations. Surprising people positioning (one user far from the group for example) can be very attracting from a saliency point of view.

On the game level, we are planning to integrate a team versus team mode. At the application start-up, the players will have to select the number of teams and the type of playground. Two types of playgrounds will be proposed: “shared” or “boxed”.

The type “shared” proposes only one playground and one goal. The gravity will not be linear but radial. The balls will fall towards the borders of the widow and reach the goal which will be located in the playground center. The players will be able to interact with other teams. The type “boxed” proposes a playground by team. No interaction with other teams will be possible. An example can be seen in Figure 11 with four teams.

The current version of the game focuses on players’ interactions. All the players having a common goal, we can track the reaction of each player against the team. In the next versions the concurrence aspect induced by the existence of several teams with different or the same goal will be interesting to analyze.

By making people play we can test in fact the social interactions between them in several contexts: in the same team with a common goal, in several teams under concurrence. The interesting aspect is that, through a game, it would be possible to observe the social evolution within a team, the emergence of a leader, the emergence of specialized players, the efficiency evolution of the team if several leaders are placed into the same team.

Finally another interesting topic is in the study of the social interactions between players who are co-located or located in different places. The first tests seem indeed to show that even if there is a common goal and the players were told to play together, people who are not located in the same place tend to act like concurrent teams and to see which location participated the most into the final structure. This is an interesting social behavior emergence which was not planned at the beginning.
8. ACKNOWLEDGMENTS

numediart is a long-term research program centered on Digital Media Arts, funded by Région Wallonne, Belgium (grant N◦716631).

The first prototype of this project has been developed during the international workshop eNTERFACE 2011 hosted in Pilsen, Czech Republic (http://enterface11.zcu.cz).

9. REFERENCES

9.1. Artistic references


9.2. Scientific references


9.3. Software and technologies


