KINterestTV - Towards Non–invasive Measure of User Interest While Watching TV

Julien Leroy, François Rocca, Matei Mancas, Radhwan Ben Madhkour, Fabien Grisard, Tomas Kliegr, Jaroslav Kuchar, Jakub Vit, Ivan Pirner, Petr Zimmermann

KINterestTV - Towards Non-invasive Measure of User Interest While Watching TV

Julien Leroy[1], François Rocca[1], Matei Mancas[1], Radhwan Ben Madhkour[1], Fabien Grisard[1], Tomas Kliegr[2], Jaroslav Kuchar[2,3], Jakub Vit[4], Ivan Pirner[4], and Petr Zimmermann[4]

[1] TCTS Lab
University of Mons, Belgium
[2] Department of Information and Knowledge Engineering
University of Economics, Prague
[3] Web Engineering Group
Faculty of Information Technology, Czech Technical University, Prague
[4] Faculty of applied science
University of West Bohemia, Pilsen
{Julien.Leroy,Francois.Rocca,Matei.Mancas,Radhwan.BenMadhkour,Fabien.Grisard}@umons.ac.be,
{Tomas.Kliegr,Jaroslav.Kuchar}@vse.cz,
{Jaroslav.Kuchar}@it.cvut.cz,
{Vit,Pirner,Zimmermann}@kky.zcu.cz

Abstract. Is it possible to determine only by observing the behavior of a user what are his interests for a media? The aim of this project is to develop an application that can detect whether or not a user is viewing a content on the TV and use this information to build the user profile and to make it evolve dynamically. Our approach is based on the use of a 3D sensor to study the movements of a user's head to make an implicit analysis of his behavior. This behavior is synchronized with the TV content (media fragments) and other user interactions (clicks, gestural interaction) to further infer viewer's interest. Our approach is tested during an experiment simulating the attention changes of a user in a scenario involving second screen (tablet) interaction, a behavior that has become common for spectators and a typical source of attention switches.

Keywords: user tracking, face detection, face direction, face tracking, visual attention, interest, TV, gesture.

1 Introduction

Imagine a 10 years old child in front of his TV today. He will probably be connected to the web, a tablet or a smartphone in his hands, to browse Wikipedia looking for additional information on the animal show that he is looking or sharing on social networks while watching the main screen.

The television, is changing to become more connected, reducing the boundary between the Internet and the television. At the age of high-speed broadband connections, the viewing experience becomes more interactive and connected to extra-content and social networks.

A recent study by Accenture [15] on consumer habits in front of their TV shows that 62% of the viewers simultaneously use a laptop while in 41% of cases a smartphone is used along with the TV. Another interesting point they highlight is that one of the main problems for the content provider is to find how to capture the customer attention by offering the right video content to respond to the viewer expectations. One of the goals of the future TV will be to be able to answer this demand. Most personalization systems are currently based on content personalization by explicit analysis of user actions (remote control, selected channels ...).

In this project and within the research program LinkedTV [1], we are interested in the explicit and implicit analysis of user actions, our goal is to design a tool for personalization based on the analysis of non verbal behavior of the viewer. To do this, the approach we used, is to analyze the attention and actions that a user can have by using a 3D sensor.

The explicit analysis is performed by the integration of both classical remote control interaction and gesture commands.

For the implicit analysis, one of the tracks that we explore is the possibility of detecting viewers' interest during the display of different media fragments on the TV screen. This information is important because it can tell us when, what and how the media interests a user, which will allow to modify the viewer profile without any explicit request or as complementary information along with explicit interactions. To achieve this goal, we implement a solution of head detection and pose estimation using a low-cost depth camera. This choice was made due to the democratization of this type of sensors and their arrival in the home through gaming platforms [24]. Moreover, TV manufacturers begin to integrate cameras into their new systems, regarding the sensors we can see the willingness of the makers to miniaturize sensors such as PrimeSense new camera "Capri" [26]. Thus, we can expect to see in the coming years 3D sensors directly integrated into televisions. But not only integrated into a TV screen, one of the interests of the video connected to the web is its availability across the network on a large number of connected components (smartphone and tablet), in which the 3D miniaturized sensors will soon also be incorporated.

Another aspect that we discussed is the ability to scan the media to assess its ability to attract the attention of a user. We want to measure the level of bottom-up attention within the images. For that a first implementation of an attention mechanism based on rarity was implemented in c++ to enable

the rapid processing of a large number of images. Based on this algorithm we proposed the concept of Metadata attention related to areas of media. Useful for understanding the behavior of a viewer may display.

The next section provides information about the technical architecture of the system which will be afterwards detailed as following: section 3 presents the explicit interaction realized with a gesture recognition method, section 4 focuses on the implicit interactions related to the attention mechanisms while section 5 details the web player and aggregator used to synchronize the viewer behavior and the displayed media fragments. Section 6 presents the experiment we realized to validate our approach and provides some cues about the analysis of the results for media personalization and finally we conclude and present future works.

2 Technical Architecture

The system, Figure 1, we have developed has five distinct modules three are integrated into a single workflow and the other 2 are additional elements supporting and adding value to the main module. The three main modules are:

1. Attention Tracker. Implicit analysis module of our system, its goal is to study the movements of a user's head and detect if he looks or not a screen and determines whether he focuses on it. The method used is based on face detection in 2D and 3D then the head pose estimation based on the resulting 3D point cloud.
2. Gesture recognition. Explicit analysis that allows the control of the interface using simple gesture recognition. The recognition technique employed uses the descriptive method by representing a simple gesture like a state machine. The gesture is recognized when all the states are validated.
3. Web interface, HTML5 player and aggregator. This module communicates with other elements using websockets, it is a web player based on Node.js [2], it can play Youtube videos while offering, by analyzing video subtitles, additional content in real time. This module will collect and aggregate all explicit actions taken by the user as well as a status of visual fixation on the player. These data is aggregated to develop a dynamic user's profile.

The other two parallel elements are:

1. Bottom-up attention metadata: the aim here is to augment the media by determining regions of interest that can provoke a bottom-up attention reaction in a subject. The image can be segmented into more or less salient regions. This information is interesting because it gives us a prediction about the probable interest of a viewer, an interest that can then be validated by the attention tracker.
2. Ground truth generator. This additional software is used for annotation and generation of database for face recognition and active appearance modeling.

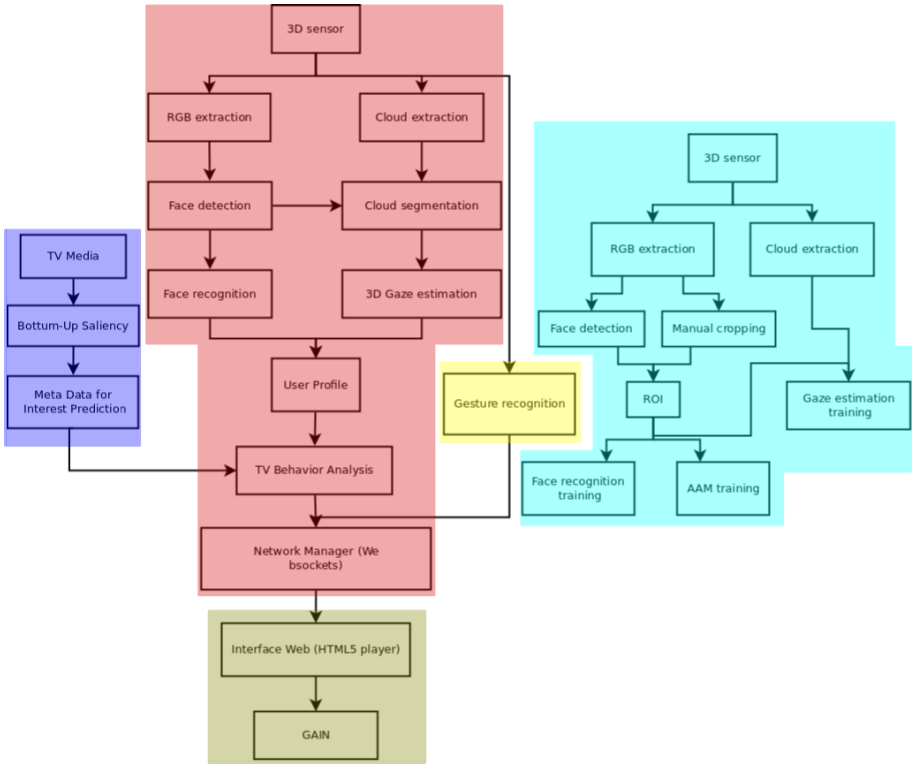

Fig. 1. The software architecture of the project with the different modules. Red: attention tracker with face recognition and head pose estimation. Yellow: gesture recognition. Blue: attention mechanism and metadata generation. Cyan: ground truth generation. Gray: player and data aggregation.

3 Explicit Interaction: Gesture Recognition

There are a lot of possible sensors which allow performing gesture recognition. On one hand, several wearable sensors are available, as accelerometers or gyroscopes. The data they provide does not need a lot of processing before using gesture recognition algorithms, and most of the time, filtering is enough. On the other hand, some sensors use standard or RGBD cameras and provide classical RGB images or/and 3D depth maps and 3D clouds of the scene. In this latter case, the acquired data has to be processed to extract the shape of the user and follow his gestures. For the LinkedTV project, we chose an RGBD sensor. This sensor provides, in addition to classical RGB images a depth map of the scene which describes the objects position related to the one of the camera. An example of depth map is displayed in Figure 2 .

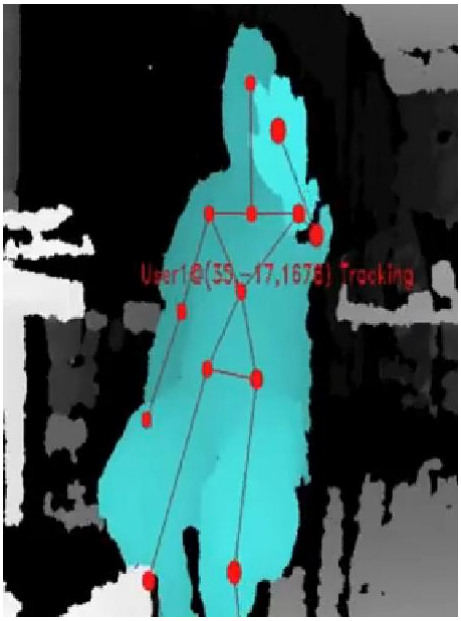

Fig. 2. RGBD camera depth map. Clear pixels are closer to the camera than dark ones. Post processing on this depth map let us extract the viewer silhouette (in cyan) and the viewer skeleton (red dots linked by red lines).

The use of an RGBD sensor is also in line with the fact that the same sensor is also used to extract interest information (see section 4). The idea is to use only one RGBD sensor to extract interest and gestures from the viewers. Currently we need to use two different sensors (one for interest and one for gestures) because the head direction extraction needs the current cameras to be not further than 1 meter from the viewer while the context camera needs a global view of the scene and this is thus located on the TV. Nevertheless, the new generation RGBD cameras (like the Microsoft Kinect 2 which will be available in 2014) will allow us to get interest and emotional cues even when the camera is far from the viewer like in the typical living rooms (2-3 meters of distance with the viewers).

For the current developments we used an Asus Xtion depth sensor which is low cost, not wearable, and is able to scan the whole scene. Furthermore, it comes with OpenNI software and the Primesens drivers that allow the find users in the scene and to track their virtual skeletons in 3D (see Figure 2). Among the lots of existing algorithms already used for gesture recognition (Gesture Follower [7], DTW [6], etc.), we chose here the simplest approach: the descriptive method. F. Kistler (from Augsburg University, Germany) developed the Full-Body Interaction Framework (FUBI) [18], an open-source framework that uses a Kinect-like RGBD sensor and that has been successfully used in many situations [20], [19], [16].

The main difference between this method and the others is the learning phase. In other approaches, we have to teach to the system how to interpret the gesture by giving it a basis of examples. With the descriptive method, the user has to learn how to perform the gesture and to do it according to the initial description. The developer defines the gestures either directly in C++ classes or in an XML file (the latter solution being more flexible as modifications can be done and reloaded while the program is running). The gestures consist in some boolean combinations of basic elements, function of the time. These basic elements are the relative position (right hand above head), orientation (left arm oriented front) or linear movements (left hand moves to the left at a minimum speed of 800mm/s) of the skeleton joints. They are updated at each newly acquired frame and give a binary outcome. These binary values are combined in different states, during a defined period of time. All the states make a kind of pipeline, and if the gesture is performed in the order of this pipeline, within the correct timings, it is detected.

Table 1. Gesture implemented and recognized in the project

Referent	**Function**	**Gesture description**
Focus	Get the system attention	Draw a circle with one hand in any direction
Play/Pause	Start to play/pause the media	The right hand stay stable 40 cm in front of the torso for at least 2 seconds
Stop	Stop to play the media	Arms crossed for at least 0.5 seconds
Next/Previous	Next/previous media/channel	Right hand moves to right/left quickly
Volume value	Set the volume value	Left arm vertical and right hand near from it, volume = 1 when the right hand is at the same height as left hand and volume = 0 when the right hand is at the same height as left elbow
Mute	Mute the volume	The left hand stay stable 40 cm in front of the torso for at least 2 seconds
Help	Pop up the help menu	Both hands near head
Add bookmark	Add a bookmark on the currently played media	Right hand stay stable 40cm in front of the torso for at least 0.3s and then moves up normally
Remove bookmark	Remove a bookmark on the currently played media	Right hand stay stable 40cm in front of the torso for at least 0.3s and then moves down normally
Lock/Unlock	Pass over controls / accept controls	Left hand above head moves left normally, then left hand above head and moves right normally

For this project, a set of 16 commonly used commands were selected, inspired by Vatavu [29]. According to Wobbrock et al. therminology [31], theses commands are called referents. This list of referents should cover all the functions

needed to control a TV in a basic use, like navigating into the media, setting the volume, interacting with menus and asking for help. They are presented in Table 1.

We opted for a limited set of referents for two reasons. The first one is the same as proposed by Vatavu [29]: "*The number of gesture commands people can remember for effective use should be limited in order not to increase cognitive load. More individual commands would translate into complex designs with a similar puzzling effect* [...] *Also, menus represent a viable option to group other, less frequent, commands*" [5]. The second one is linked to the gesture recognition method we use: more gestures could lead to interaction between them and unwanted detections.

To limit the interactions between gestures, we added a "focus" command, a gesture to be performed before most of the other commands to get the attention of the system. If no gestures have been detected after 2.5 seconds, the focus is lost and all the new gestures are ignored until the focus gesture is performed. The TV can be locked or unlocked to prevent any gesture performed in front of the system to be interpreted as a command. It is the same idea as Focus command but in a more restrictive way. Only gestures which need to be done immediately like bookmark do not need to be initiated using the focus gesture.

For flexibility reasons, the gesture description has been implemented in an XML file. Most of the gestures were inspired by [28]. In this experiment, people were told to imagine gestures to match each referent, although the referents were not exactly the same as in our case. After some experiments we agreed on this set of gestures, some of them are used for different referents, depending on the context.

According to FUBI implementation, there are different types of gestures. Postures are static gestures that have to be maintained for a certain period of time to be detected. Linear movement are a simple movement performed at a certain speed (we chose 1m/s for normal speed and 2m/s for fast speed). Combinations are complex gestures which need more than one linear movement to be described. Dynamic postures are like postures but one of the joint is moving and its position, relatively to other joints, is translated into a continuous value (e.g. to control a continuous parameter, such as volume).

As it will be used very often, the focus gesture should be easy to remember and to perform. We chose to implement it as a circle, drawn with the right or the left hand in any direction. The only restriction is to start it from the top.

Each time a gesture is recognized, a message is sent to the attention tracker system (see section 4). The attention tracker packs the message by using the websockets protocol and sends it to the web player (section 5) which is controlled by those gestures. Some controls (play/pause, etc) are fattened by the web player with the video or media fragment ID and time and forwarded to the aggregator system (section 5.2).

4 Implicit Interaction: Attention Tracker

Movement and orientation of the head are important non-verbal cues that can convey rich information about a person's behaviour and attention [30][17]. Ideally, to find out if a user is looking at the screen or not, we should extract the ocular movements of the subject. But given the experimental conditions mainly in terms of sensor to viewer distance and in terms of sensor resolution, it is not possible for us to have access to such information. Therefore our system will be based on the assumption that to detect changes in visual focus, the gaze of a person is considered to be similar to the direction of his head. As stated in [25], "[...]*Head pose estimation is intrinsically linked with visuel gaze estimation ... By itself, head pose provides a coarse indication of gaze that can be estimated in situations when the eyes of a person are not visible*[...]". Several studies rely and validate this hypothesis as shown in [3]. Therefore, we will detect visual attention switches and focus by studying the orientation of the head.

Until recently, the literature has mainly focused on the automatic estimation of the poses based on standard images or videos. One of the major issues that must be addressed to obtain a good estimator is to be invariant to variables such as: camera distortions, illumination, face shape and expressions or features (glasses, beard). Many techniques have been developed over the years such as appearance template methods, detector array methods, non linear array methods, manifold regression methods, flexible methods, geometric method, tracking method and hybrid methods. More information on these methods can be found in [25]. More recently, with the arrival of low cost depth sensor, more accurate solutions have emerged [12][10]. Based on the use of depth maps, those methods are able to overcome known problems on 2D images as illumination or low contrast backgrounds. In addition, they greatly simplify the spatial positioning of the head with a global coordinate system directly related to the metric of the analysed scene. Many of these techniques are based on a head tracking method which unfortunately often requires initialization and also undergoes a drift. Another approach, based on the frame to frame analysis as the method developed by [11], provides robust and impressive results. This method is well suited for a living room and TV scenario. It is robust to illumination conditions that can be very variable in this case (dim light, television only source of light, etc.) but is based on a 3D sensor like the Microsoft Kinect.

The approach we propose here is based on the work developed in [23] and [22]. To improve the exploitation and use of our system as an element to be integrated into a set top box, the system architecture and interaction of different elements have been integrated as in shown in Figure.3.

The proposed system is based on the head detection and pose estimation on a depth map. Our goal is to achieve head tracking in real time and estimate the six degrees of freedom (6DOF) of the detected head (spatial coordinates, pitch, yaw and roll). The advantage of a 3D system is that it uses only geometric information on the point cloud and is independent of the illumination issues which can dramatically change in front of a device like a TV. The proposed system can even operate in the dark or in rapidly varying light conditions, which

is not possible with face tracking systems working on RGB images. In addition, the use of 3D data provide more stable results than 2D data which can be mislead by projections of the 3D world on 2D images. Finally, the use of depth maps let us extract people position and features. This is also important as people detection with no face detection means that the head either has a more than 60 degrees of pitch or 75 degrees of yaw.

The method used here is based on the approach developed in [13][14] and implemented in the PCL library [4]. This solution relies on the use of a random forest [8] extended by a regression step. This allows us to detect faces and their orientations on the depth map. The method consists of a training stage during which we build the random forest and an on-line detection stage where the patches extracted from the current frame are classified using the trained forest.

The training process is done only once and it is not user-dependent. One initial training is enough to handle multiple users without any additional configuration or re-training. This is convenient in a large public setup as the one of people watching TV. The training stage is based on the BIWI dataset [14] containing over 15000 images of 20 people (6 females and 14 males). This dataset covers a large set of head pose ($\pm$75 degrees yaw and $\pm$60 degrees pitch) and generalizes the detection step.

During the test step, a leaf of the trees composing the forest stores the ratio of face patches that arrived to it during training as well as two multi-variate Gaussian distributions voting for the location and orientation of the head. This step of the algorithm provides the head position and a rough head orientation on any new individual without the need of re-training. We than apply a final processing step which consists in registering a generic face cloud over the region corresponding to the estimated position of the head. This last step greatly stabilizes the final head position result.

To improve the performance of tracking and include elements such as face recognition, the major change that we made on the software architecture was to use a 2D face detection (HAAR) as a pre-filter step, Figure 3. This first step performed on the RGB image from the sensor has several advantages:

1. Information limitation. It reduces the cloud information that need to be processed for estimating the users head orientation: the classification of the underlying point cloud can be speeded up.
2. Cross detection. The 2D face detection has also the other advantage to be a predetection test and eliminates some false detections which might occur if the system was only based on the geometrical data.
3. Face recognition. Based on the face detection, we realize a face recognition step to identify the user. This information is used to recognize a known user and to track his beahavior. The face recognition process work by fusing the results of 3 classical face recognition algorithms implemented in the OpenCV library (LBPH, FisherFace, EigenFace).

To detect if a user watches the screen or not, we reconstruct a virtual simplified model of the real scene. Therefore, knowing the 6DOF position of the face of

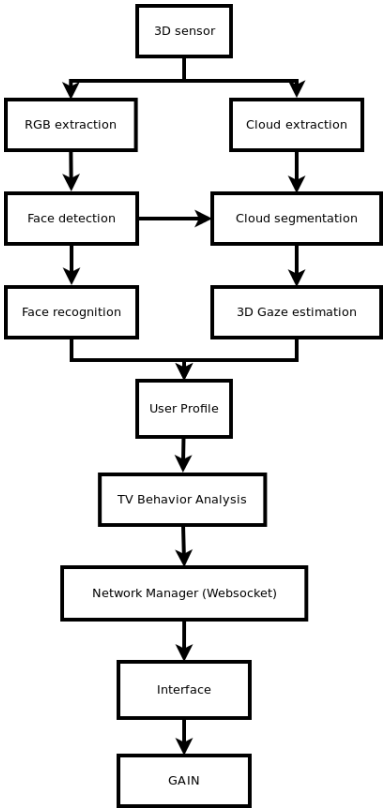

Fig. 3. Attention tracker workflow: 3D head pose and face recognition result go to the user profile and TV behaviour analysis which proceed to information low-level processing and fusion and forward it to the network manager module. The network manager takes all the messages (from the interest module, context tracking module and gesture module) and sends them to the player using the websockets protocol. The player enriches the messages with the video ID and time and forwards to the GAIN module that will aggregate the data.

the person detected, it is possible to estimate the point of intersection between the screen virtual model and the orientation of the head (Figure 4). In this way, we can synchronize annotated media with the head tracker and estimate where the user is looking.

This information is sent to a user manager where it is fused with gestural information (obtained as described in section 3) and than forwarded to the network manager module which sends it using websockets protocol to the web player (section 5).

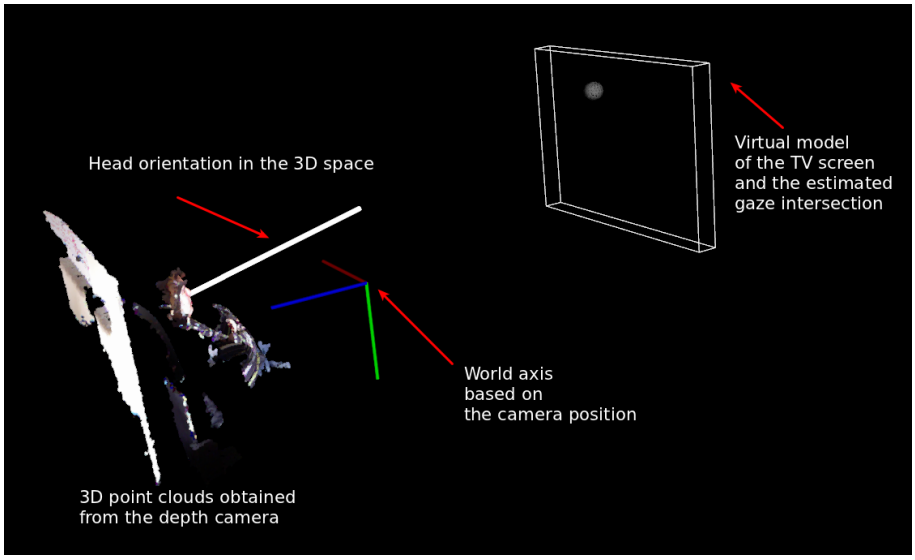

Fig. 4. 3D rendering of our system. On the left, we can observe the 3D point cloud obtained with the depth camera. The head pose estimation algorithm is applied on this cloud, if a face is detected, we retrieve a vector of the head direction and compute an estimation of where the user is watching on the virtual screen.

4.1 MetAttention: Image Metadata Based on a Visual Attention Mechanism

If it is possible to identify where a user approximatively looks at, this information can be supplemented by bottom-up attention induced by the media. To investigate this kind of attention and couple it with the observation done on the user's behavior, we implemented a computational attention mechanism to analyze the bottom-up stimuli sent by the media. This algorithm is based on a bottom-up attention mechanism using a multi-scale rarity [27].

There are three main steps. First, we extract low-level colour and medium-level orientation features. Afterwards, a multi-scale rarity mechanism is applied. Finally, we fuse rarity maps into a single final saliency map. Contrary to RGB color space, some alternative colour spaces (in our case YCbCr) give better uncorrelate colour information. Moreover, the nonlinear relations between their component are intended to mimic the nonlinear response of the eye. At this stage, the algorithm split in two pathways. The first one, mainly deals with colours (low-level features) while the second one with textures (medium-level features). While the first pathway directly uses the colour transformation and computes its rarity, the second pathway extracts orientation features maps by using a set of Gabor filters. These filters were chosen because they are similar to simple cells of the visual cortex in the brain [9]. For more information about the

attention mechanism which was partly implemented, the reader can refer to [27]. The algorithm was implemented in C++ using OpenCV and multithreading, the performance in comparison of the Matlab implementation is 10x.

The output of the algorithm provides us with a map of saliency, using different steps of filtering and morphological operations it is then possible to segment the image into areas with high saliency values by using an adaptive thresholding. These areas will allow us to generate regions of bottom-up interest we can therefore correlate with measures of user's head direction obtained using the attention tracker.

Figure 5 shows the main processing steps of the algorithm:

1. The original image is converted in the YCbCr color space;
2. On each color channel a set of Gabor filters are applied to extract direction information and after that a multi scale rarity mechanism is applied;
3. The 6 rarity maps are fused together into a single map. This fusion is achieved in two main steps: an intra-channel fusion followed by an inter-channel one. The result is called a saliency map.

5 TV Web Player and Aggregator

Figure 6 depicts the simplified workflow and communication of the modules at different levels. There are three levels: Web Browser, Server, Sensors. The application on the Sensors level was is implemented in C++ and communicates to the server using Websocket protocol (sections 3 and 4). The server is implemented in Node.js and the application in the browser is implemented in HTML and JavaScript that communicates with the server using the Websocket protocol and REST API.

5.1 Player

The Player simulates the Smart TV environment using videos from YouTube (Figure 7). It is implemented as a web based application within a web browser. The interface provides the main screen with the video player, basic controls of the player and a semantic description of content based on analysis of subtitles. The viewer can interact with video using basic controls buttons or the user can read related content by clicking on the links to find more related information. Both Player and sensors for attention tracking and sensors for gesture control are connected using Websocket protocol. All detected gestures and interest clues detected by the attention tracker are sent to the synchronization service on the server. This information is propagated in nearly real-time to the Player. The player translates the incoming message into proper actions. Gestures control the player and attention information change is complemented with the video ID and video time which is displayed at the moment where the attention change occurs. All of these interactions, including actions from sensors, are sent to the GAIN component (section 5.2) using a REST interface.

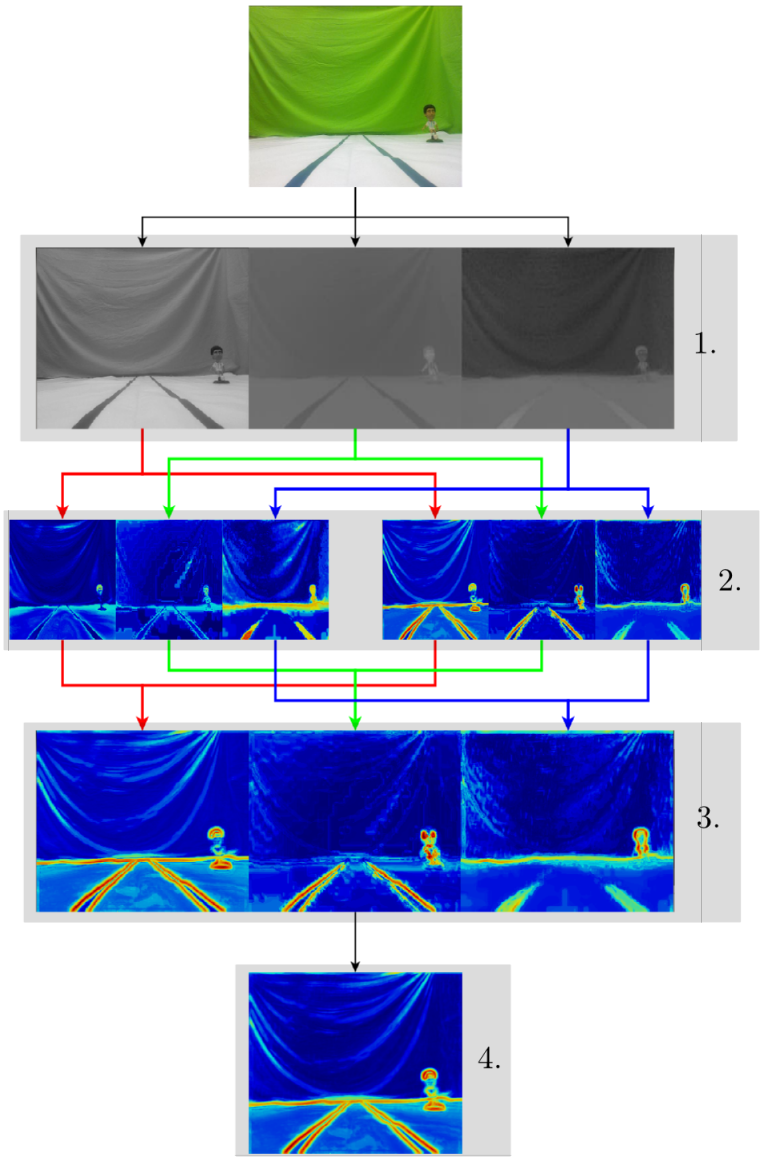

Fig. 5. Diagram of our used model. First, from the input image, a change of color space is applied to improve the color differentiation (1.). Colour and orientation features are extracted. Then, for each feature, a multi-scale rarity mechanism is applied (2.). Finally, two fusions (intra- and inter-channel) (3.) are made from the rarity maps to provide the final saliency map (4.).

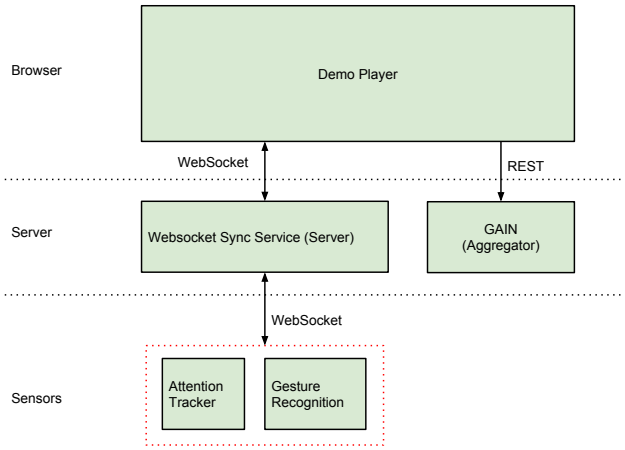

Fig. 6. Workflow

Fig. 7. An extended version of the YouTube player used for the SmartTV demo

5.2 GAIN - General Analytics INterceptor

GAIN (www.inbeat.eu) is a web application and service for capturing and preprocessing user interactions with semantically described content. GAIN outputs a set of instances in tabular form suitable for further processing with generic machine-learning algorithms. GAIN is implemented in Node.js and it is composed of three modules. First, a tracking module is responsible for capturing information. Afterwards, the storage module accumulates all data within a database. Finally, the aggregation module process the data and provides the outputs. GAIN provides RESTful API for collecting information and for aggregated outputs. GAIN will provide a timeline of the displayed video containing all synchronized actions which occurred during the viewer TV experience. The syn-

chronized data contain both explicit (user actions like clicks, play, stop, etc.) and implicit information (user behavior observation through the attention tracker). Examples of the output data can be seen in figures such as Figure 10.

6 Experimentation

To validate our approach of change detection in the viewer focus of attention, we designed an experiment leading to a moment of focus on a specific period of the broadcast media. Our goal was to determine whether it was possible for us to detect this moment through our system, but also to observe whether different behaviors stood out in conducting the experiment.

The experience consisted in watching 4 different videos in English and subtitled, through our web player (section 5). Each video has a different content and dealt with topics like politics or sports. While viewing these videos, additional content is provided to the user in real time on the web player (Figure 7 on the right). The subject were asked to be soccer fans and we asked them to watch the moments dedicated to this topic on the different videos which were displayed on the web player. To check that they really pay attention to soccer information, the users were asked to answer a questionnaire about relevant videos. Finally, to also cause changes in the individual attention we asked them to fulfil two different tasks while viewing the media:

1. The first task was to play a simple puzzle game on a tablet (IPad) and get the maximum possible score within the time limit.
2. The second task was to answer a series of questions on soccer dedicated to the broadcast videos. The stimulated interest was soccer and concerned only videos 2 and 4. Some questions had a higher difficulty push the user to exploit the extended content (access to Wikipedia which is available on the web player on the right side of Figure 7) to correctly answer the questionnaire.

6.1 Course of the Experiment

Our experiment took place in two stages. First, given the complexity of the operations to be performed, a tutorial simulating the experience is proposed to the subject. He could for 4 minutes try a simulation of the experience and learn how to handle the different elements that are provided (play/pause interactions, questionnaire and web player). At the end of this training phase, the actual experiment was performed with a time limit of 7 minutes. When the experiment was completed, the system stops and we collect the questionnaire and the game score.

The displayed media order is as follows:

1. The first video (2 min 18) is about a new US and South Korea joint defense plan against any provocation from North Korea and the help the US can provide to his ally. (url: `https://www.youtube.com/watch?feature=player_`

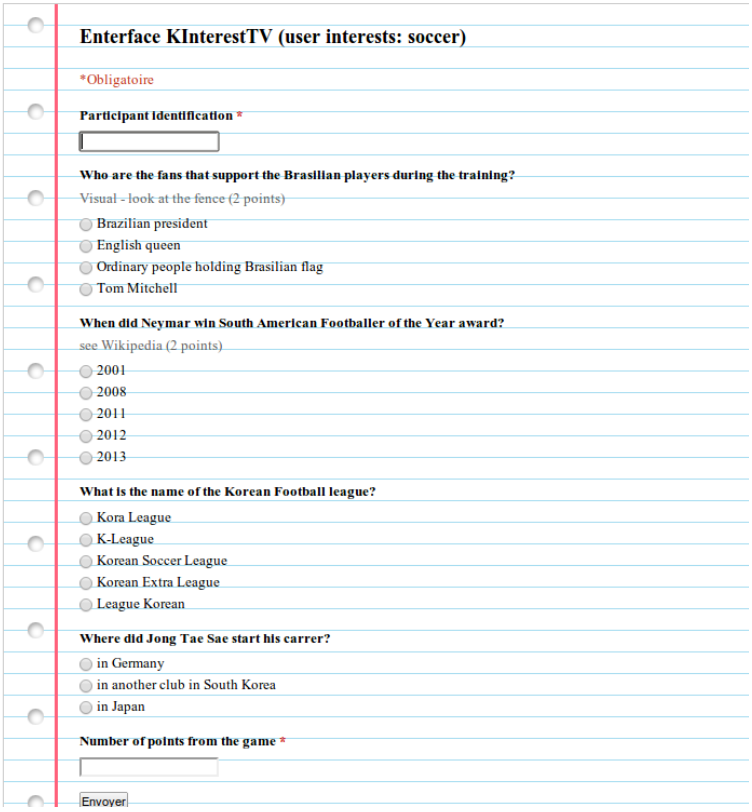

Fig. 8. Questions given to the user. The stimulated interest here is about soccer.

`embedded&v=k4JstBdOsgk`). This video has no information about the user simulated interest (soccer).

2. The second video (1 min 03) is about soccer, precisely about an England versus Brazil match and gives information about important player in the teams. (url: `https://www.youtube.com/watch?feature=player_detailpage&v=do5NcLT-t3s`)

3. The third video (1 min 08) is about tourism and the 10 top attractions in Berlin. (url: `https://www.youtube.com/watch?feature=player_embedded&v=f9Uxzvekgio`)? Again, this video provides no information on the user simulated interest.

4. The fourth video (1 min 03) is about soccer and the transfer of a North Korean striker to South Korean club. (urel: `https://www.youtube.com/watch?feature=player_detailpage&v=vO4ZM8HG4yg`)

Fig. 9. Experimental setup: videos are displayed on a computer screen while the user needs to play a game and answer questions simultaneously. A 3D sensor on top of the screen captures the head movement of the user.

6.2 Interest Beats and Results

At the end of the experiment, 10 people have used our sytem and responded to the questionnaire. Based on the attention data obtained by our system, we generated a visualization of actions taken by the user that we call the "Interest beat". The expected behavior was the following and can be observed on the Figure 10:

1. While the videos not related to the questionnaire are playing (1st and 3rd), the user will mainly focus on the game on the second screen (and therefore will not look ot the main screen)
2. While the videos related to the questionnaire are playing, the viewer looks to the main screen and therefore stops playing the game on the second screen. He can also stop or jump in this video to go back to a topic related to the questions of their questionnaire

The interest beat is a graph presenting the output of the GAIN module [21] and consists in time synchronized events happening all over the content (X axis) for the main (recognized) viewer. From top to down (Y axis) we can see the different events (click on links, pause, play, interest off main screen, interest on main screen and no information about viewer interest). The latest category means that the face of the user is not visible anymore and the head pose is not available anymore (out of the range of +/-75 degrees yaw and +/-60 degrees pitch. The example of a viewer in Figure 10 summarized the typical viewer behaviour:

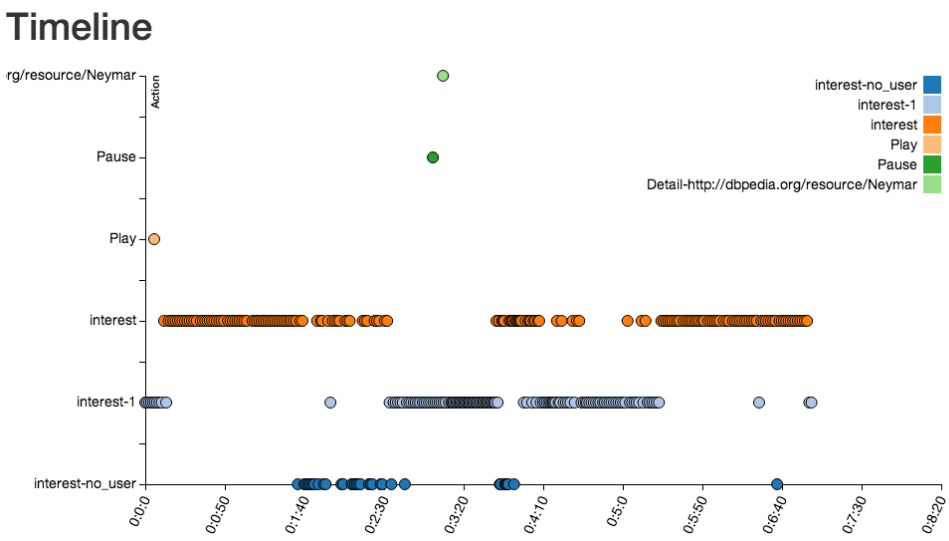

Fig. 10. Interest beat. The timeline comes from each event while the experiment. It represents the viewer behaviour over the displayed content (first video from 00:00:00 to 00:02:18, second video until 00:03:21 and the third video until the end 00:04:24). On the Y axis: clicks on links, pause, play, interest-0 (not looking at the main screen), interest -1 (looking at the main screen) and interest-no-user (user not found anymore).

1. First he clicks on play ...
2. In the same time he watches the content during some seconds (called "Interest-1" on the Y axis)
3. When he realizes that the first video is not related to the questionnaire he does not look anymore to the main screen and begins to play the game on the second screen (tablet)
4. When the second video (which contains the answers to his questionnaire) begins, he focuses at the main screen again
5. At some point he uses pause and play explicit actions to have time to answer the questions of his questionnaire. He also clicks on a link provided by the viewer as one of the questions requires the user to get extra information using this link.
6. Once the second video is finished, he mainly focuses again on his tablet game and not on the main screen.

On the ten users the system managed to follow the head direction of 9 participants while it failed on one of them. While the explicit actions (play/pause/click on link, etc.) were performed, the other users have a significant activity towards the main screen. Nevertheless, it is not easy to asses the tracking performance as the users sometimes have different behavior. Sometimes the viewer forgot to answer the questions, so he had to go back in the video to do it at the end like in Figure 11.

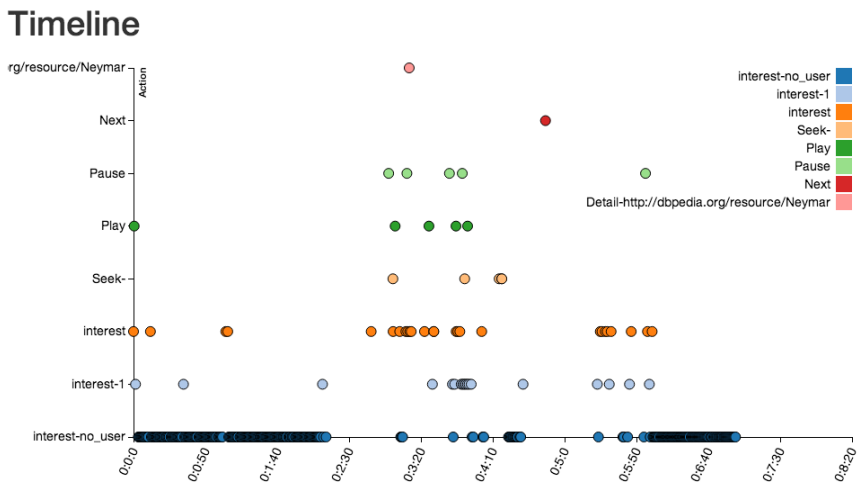

Fig. 11. Interest beat with a user looking at the predicted video but has forgotten to interact to answer the question.

In other users, they are very consistent with the task (playing or looking to the screen like in Figure 10 while other alternate much more the gaze between the main screen and the game like in Figure 12. On the nine viewers where the tracking worked, the results are consistent with the scenario and encouraging.

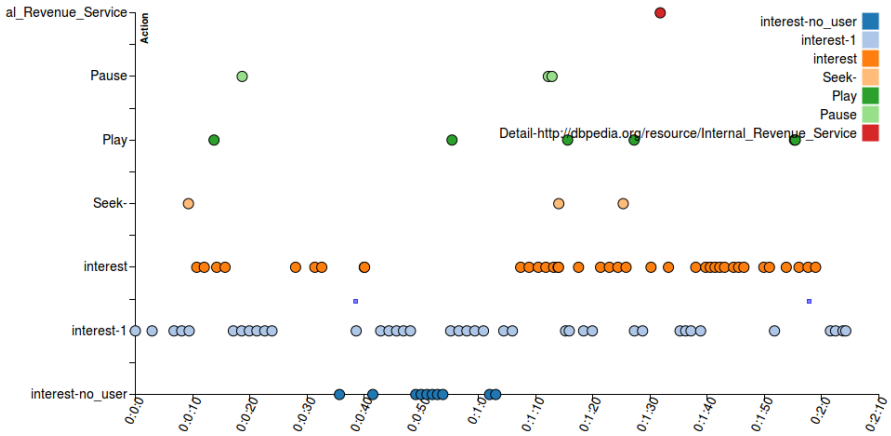

Fig. 12. Interest beat with a user alternatively looking at the main screen and at the game.

7 Conclusion

In this paper, we presented the whole architecture of our implicit behavior analysis system based on a 3D head tracker and also the explicit gesture recognition system. We also describe the additional information which can be provided by the extraction of low-level bottom-up features from the media. The results show that it is possible to extract implicit information in an efficient and consistent way on where and when people look at their TV. Modules which provide a web player and a data aggregator are used to synchronized all the behavioral analysis of the viewer. This work is designed to further feed a personalization framework capable of processing behavioral data to dynamically enhance the profile of a user. This profile change needs further machine learning algorithms which take the synchronized data of the proposed system as an input and process it in order to obtain the user interest for the different media fragments and media links which were shown to the user.

8 Acknowledgments

This work is supported by the Integrated Project LinkedTV (www.linkedtv.eu) funded by the European Commission through the 7th Framework Programme (FP7-287911).

References

1. Linkedtv project. http://www.linkedtv.eu
2. Node js. http://nodejs.org/
3. Abe, K., Makikawa, M.: Spatial setting of visual attention and its appearance in head-movement. IFMBE Proceedings 25/4, 1063–1066 (2010), http://dx.doi.org/10.1007/978-3-642-03882-2_283
4. Aldoma, A.: 3d face detection and pose estimation in pcl (September 2012)
5. Bailly, G., Vo, D.B., Lecolinet, E., Guiard, Y.: Gesture-aware remote controls: guidelines and interaction technique. In: Proceedings of the 13th international conference on multimodal interfaces. pp. 263–270. ICMI '11, ACM, New York, NY, USA (2011), http://doi.acm.org/10.1145/2070481.2070530
6. Bettens, F., Todoroff, T.: Real-time dtw-based gesture recognition external object for max/msp and puredata. In: in Proc. SMC 09, 2009. pp. 30–35
7. Bevilacqua, F., Guédy, F., Schnell, N., Fléty, E., Leroy, N.: Wireless sensor interface and gesture-follower for music pedagogy. In: Proceedings of the 7th international conference on New interfaces for musical expression. pp. 124–129. NIME '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1279740.1279762
8. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001), http://dx.doi.org/10.1023/A%3A1010933404324
9. Daugman, J.G.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. J. Opt. Soc. Am. A 2(7), 1160–1169 (Jul 1985), http://josaa.osa.org/abstract.cfm?URI=josaa-2-7-1160

10. Fanelli, G., Gall, J., Van Gool, L.: Real time head pose estimation with random regression forests. Cvpr 2011 pp. 617–624 (Jun 2011), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5995458
11. Fanelli, G., Gall, J., Van Gool, L.: Real time 3d head pose estimation: Recent achievements and future challenges. In: Communications Control and Signal Processing (ISCCSP), 2012 5th International Symposium on. pp. 1–4 (2012)
12. Fanelli, G., Dantone, M., Gall, J., Fossati, A., Gool, L.: Random Forests for Real Time 3D Face Analysis. International Journal of Computer Vision 101(3), 437–458 (Aug 2012), http://link.springer.com/10.1007/s11263-012-0549-0
13. Fanelli, G., Dantone, M., Gall, J., Fossati, A., Gool, L.: Random forests for real time 3d face analysis. International Journal of Computer Vision 101, 437–458 (2013), http://dx.doi.org/10.1007/s11263-012-0549-0
14. Fanelli, G., Weise, T., Gall, J., Gool, L.V.: Real time head pose estimation from consumer depth cameras. In: Proceedings of the 33rd international conference on Pattern recognition. pp. 101–110. DAGM'11, Springer-Verlag, Berlin, Heidelberg (2011), http://dl.acm.org/citation.cfm?id=2039976.2039988
15. Francesco Venturini, Charles Marshall, E.D.A.: Hearts, minds and wallets winning the battle for consumer trust accenture video-over-internet consumer survey (2012)
16. Frisson, C., Keyaerts, G., Grisard, F., Dupont, S., Ravet, T., Zajga, F., Colmenares-Guerra, L., Todoroff, T., Dutoit, T.: Mashtacycle: on-stage improvised audio collage by contentbased similarity and gesture recognition. In: 5th International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN) (2013)
17. Gaschler, A., Jentzsch, S., Giuliani, M., Huth, K., de Ruiter, J., Knoll, A.: Social behavior recognition using body posture and head pose for human-robot interaction. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems pp. 2128–2133 (Oct 2012), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6385460
18. Kistler, F.: Fubi- full body interaction framework (2011), http://www.informatik.uni-augsburg.de/lehrstuehle/hcm/projects/tools/fubi/
19. Kistler, F., Endrass, B., Damian, I., Dang, C., Andr, E.: Natural interaction with culturally adaptive virtual characters. Journal on Multimodal User Interfaces pp. 1–9, http://dx.doi.org/10.1007/s12193-011-0087-z, 10.1007/s12193-011-0087-z
20. Kistler, F., Sollfrank, D., Bee, N., André, E.: Full body gestures enhancing a game book for interactive story telling. In: Proceedings of the 4th international conference on Interactive Digital Storytelling. pp. 207–218. ICIDS'11, Springer-Verlag, Berlin, Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-25289-1_23
21. Kuchař, J., Kliegr, T.: Gain: web service for user tracking and preference learning - a smart tv use case. In: Proceedings of the 7th ACM conference on Recommender systems. pp. 467–468. RecSys '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2507157.2508217
22. Leroy, J., Rocca, F., Mancas, M., Gosselin, B.: 3d head pose estimation for tv setups. In: Proceeding of Intetain2013: 5th International Conference on Intelligent Technologies for Interactive Entertainment (2013)
23. Leroy, J., Rocca, F., Mancas, M., Gosselin, B.: Second screen interaction: an approach to infer tv watcher's interest using 3d head pose estimation. In: Proceedings of the 22nd international conference on World Wide Web companion. pp. 465–468. WWW '13 Companion, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2013)

24. Microsoft: Kinect sensor. http://www.xbox.com/kinect
25. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: a survey. IEEE transactions on pattern analysis and machine intelligence 31(4), 607–26 (Apr 2009), http://www.ncbi.nlm.nih.gov/pubmed/19229078
26. PrimeSense: Capri sensor. http://www.primesense.com/news/primesense-unveils-capri
27. Riche, N., Mancas, M., Duvinage, M., Mibulumukini, M., Gosselin, B., Dutoit, T.: Rare2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis. Signal Processing: Image Communication 28(6), 642 – 658 (2013), http://www.sciencedirect.com/science/article/pii/S0923596513000489
28. Vatavu, R.: A comparative study of user-defined handheld vs. freehand gestures for home entertainment environments. Journal of Ambient Intelligence and Smart Environments
29. Vatavu, R.D.: User-defined gestures for free-hand tv control. In: Proceedings of the 10th European conference on Interactive tv and video. pp. 45–48. EuroiTV '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2325616.2325626
30. Vinciarelli, A., Pantic, M., Bourlard, H.: Social signal processing: Survey of an emerging domain. Image and Vision Computing 27(12), 1743 – 1759 (2009), http://www.sciencedirect.com/science/article/pii/S0262885608002485
31. Wobbrock, J.O., Morris, M.R., Wilson, A.D.: User-defined gestures for surface computing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1083–1092. CHI '09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1518701.1518866