An immersive view approach by secure interactive multimedia proof-of-concept implementation

Pablo Antón · Antonio Maña · Antonio Muñoz · Hristo Koshutanski

Abstract Live media streaming is a field that recently has had a great impact in the scientific community, especially in the case of interactive media streaming. In this paper we propose a reference architecture conceptualizing an immersive view effect by considering heterogeneous tracking devices and enriched movement control over heterogeneous stream image sources. A proof-of-concept prototype implementation of the reference architecture is presented, called Live Interactive FramE (LIFE), illustrating the potential and value of the immersive view concept. Our work is part of the DESEOS research project that aims at applying an Ambient Assisted Living paradigm to a targeted scenario of hospitalised children. The main goal of LIFE is reducing stress and isolation during hospitalisation by enabling an immersive view of school activities via live media streaming. Functional and security aspects of LIFE are presented along with details of implementation and performance evaluation. Conclusions of experiments show that LIFE enables practical secure media streaming solution with optimal video quality settings.

Keywords Virtual window · Immersive view · Multimedia security · Ambient assisted living

P. Antón (✉) · A. Maña · A. Muñoz · H. Koshutanski
E.T.S.I. Informatica, Campus de Teatinos, Malaga 29071, Spain
e-mail: panton@lcc.uma.es

A. Maña
e-mail: amg@lcc.uma.es

A. Muñoz
e-mail: amunoz@lcc.uma.es

H. Koshutanski
e-mail: hristo@lcc.uma.es

1 Introduction

The ever-increasing bandwidth of the Internet and the adoption of new technologies (e.g. NGN [10] and SIP [24], RTP [25], etc.) related to the transmission of multimedia signals over IP networks have fostered the emergence of many interesting video-based applications. However, while the use of Internet for this purpose obviously opens many possibilities for interactivity, few systems have really exploited this feature, and the interactive capabilities have remained very limited and applied only to the selection of a few preferences by the user. Likewise interactive Internet-based TV has been advertised for years without going much further than the provision of some user feedback.

In the field of communications, and in multimedia in particular, the concept of control refers to the control of the low-level communication mechanism and protocol, thus dealing with issues like quality of service, error management and reporting, communication establishment and configuration, client management, etc. However, in this paper we are interested in control in the sense of streaming media control from the user perspective, thus focusing on aspects related to source control (pan, tilt, zoom, camera movement), transport control (play, pause), etc.

In some fields, such as video-surveillance or teleconferencing, some capabilities for interactive control of the image have been offered, but these capabilities have been developed in an ad-hoc and proprietary manner and have not been standardized, thus representing only a very limited advance.

In this paper, we present an architecture that supports advanced image control for the transmission of multimedia streaming. The architecture uses independent streaming and control channels to enable advanced control capabilities. We have implemented the architecture in a project, called DESEOS [2], in the domain of Ambien-Assisted Living (AAL) [8]. This project has the goal of developing systems to help children who spend long periods in hospitals to keep in touch to their family and school environments. One of those systems is the Live Interactive FramE (LIFE); a system that simulates a virtual window to a remote location with the goal of providing an immersive effect. The goal of LIFE is to increase the feeling of being present in the classroom, thus reducing stress and increasing the school performance of the children being cared for. LIFE requires a special type of interactive media streaming service in order to simulate the effect of looking through a window by changing the part of the classroom displayed depending on the position and viewing angle of the hospitalised child.

In the rest of the paper, Section 4 overviews related approaches and positions them with respect to LIFE. Section 2 describes the general reference architecture for the interactive multimedia streaming. Section 3 overviews the LIFE application scenario, as well as requirements and implementation details of the LIFE application together with performance evaluation results. Section 5 concludes the paper, and Section 6 outlines future work.

2 Our proposal

Our approach aims at generalizing the concepts of a virtual window and a remote media space control by providing support for (i) heterogeneous mechanisms

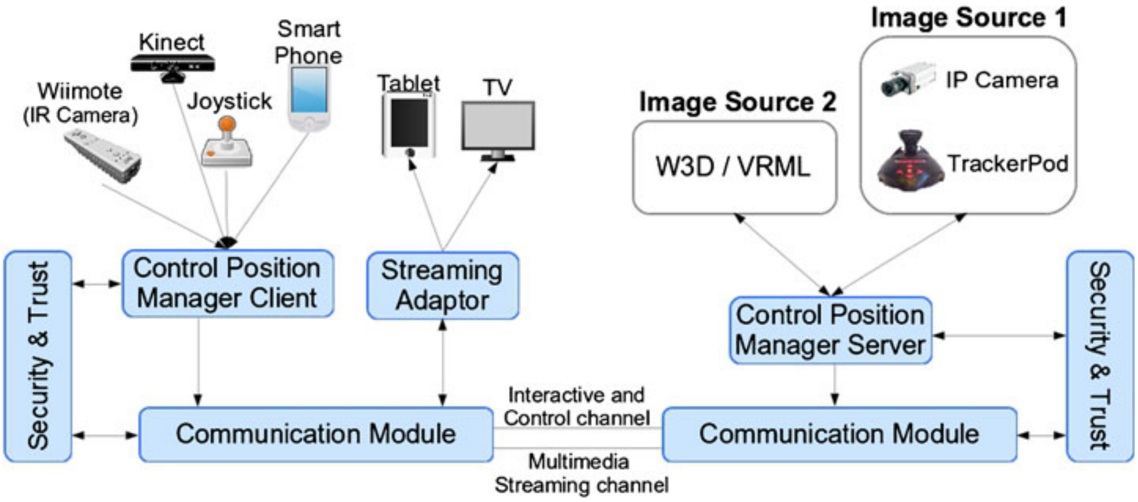


Fig. 1 Immersive view reference architecture

representing (tracking or interpreting) user's position or movement intention to control remote media source; and (ii) enriching movement control over heterogeneous stream image sources, for example real-time streaming or 3D interactive environments.

The reason for the heterogeneous device support is being able to provide from a more economic solution e.g., using wearable glasses, to more flexible but expensive devices, e.g., Kinect.[1] The reason for supporting movement control over heterogeneous image sources is to enable adoption of our approach in a wider set of application domains.

We propose an architecture designed to take into account a variety of devices to enable tracking of user's position, and a variety of image sources to enable immersive view. Figure 1 shows the conceptual view of the architecture. The architecture enables certain level of interactivity on the video streaming (regardless of the image source). The observer is an active participant in the communication, in such a way that the system deduces the remote perspective desired by the observer. The architecture is based on a bidirectional client-server model, and is composed by the following modules:

- *Control position manager*. The Control Position Manager Client (CPMC) component is in charge of deducing the perspective that the observer wants to visualize (e.g.., Wii based head tracking described in Section 4.1). A relevant aspect of the CPMC consists of its design, which is devised to work with heterogeneous devices (e.g., wiimote, Kinect, joystick, smart phone, etc). CPMC translates the desired perspective into data that are the basis to compute the new coordinates in the desired perspective. The Control Position Manager Server (CPMS) component gets as input the information of the desired perspective coordinates from the CPMC and computes these data in the remote space to provide the new perspective (i.e., spinning the camera to a specific target position). CPMS is designed to work with different image sources (i.e., IP camera, trackerpod, pre-recorded Video, etc).

[1] http://www.xbox.com/es-es/kinect

- *Communication Module*. This component deals with the creation and configuration of the streaming and control channels. In the configuration process, several security aspects and vision ranges can be handled. As it is shown in Fig. 1, two instances of the Communication Module are connected through two different channels:
    - *Interactive and control channel*. Data transmitted through this channel are used for configuration and communication of coordinates with the remote space.
    - *Multimedia streaming channel*. This channel is used for the streaming transmission. A catalogue of different protocols can be implemented in this channel (i.e., SRTP, RTMP, VRTP, etc.).
- *Streaming Adaptor Module (SAM)*. An optional component that is in charge of changing some streaming conditions on the client side according to some context conditions (i.e., to resize the frame for a natural vision in low bandwidth connections).
- *Security and Trust Manager*. An optional component that is in charge of (i) enabling secure streaming between the two communication sides on both channels, including trusted channel establishment, (ii) enabling security on the level of device authentication and image source secure identification, and (iii) enabling controlled access to CPMS by observer's side CPMC component.

The proposed model can be seen as a reference architecture targeting immersive view effect over an image source. The remote media space control is achieved by tracking/interpreting location and position aspects of the observer instead of the observed object. The actual realization and implementation of the architecture will depend on the application domain and related scenarios, as we will see in the rest of the paper.

3 LIFE implementation

As a proof of concept implementation, called LIFE, has been developed for the provision of a set of means to mitigate problems caused by long-term hospitalisation of children [20, 22].

3.1 Application scenario

We consider a scenario which involves two spaces (physical environments): Hospital where children are hospitalised; School where the hospitalised children's classmates and teacher stay. Figure 2 illustrates the set of devices considered in each of these spaces described as following:

- School: A video capture device (IP camera or webcam), a microphone (if not integrated in the camera), and a PTZ (Pan, Tilt, Zoom) device (TrackerPod[2]).
- Hospital: An information displayer (monitor or TV screen), and a head tracking device such as the bluetooth infrared camera of Nintendo Wii remote control or a Kinect [13, 15].

[2] http://www.trackercam.com/TCamWeb/productdes.htm

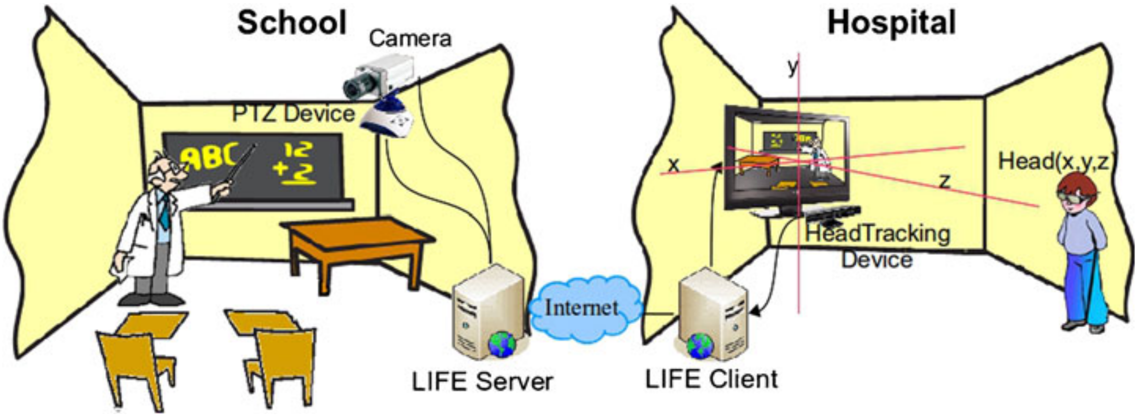


Fig. 2 LIFE application scenario

3.2 Scenario requirements

The scenario described above entails a minimum set of functional requirements of the LIFE application such as:

- *Fluent video streaming*: it is essential to achieve a fluent video streaming that produces a natural reality feeling for the user.
- *Accurate and instantaneous user tracking*: it is important to achieve smooth control on the remote media space by computing an accurate and instantaneous user position.
- *Zoom capabilities*: it is essential to provide zoom capabilities on the remote image source to achieve real feeling of the immersive view effect.

Given the target users of LIFE application are hospitalized children, security and privacy aspects are important to be addressed. We have identified several specific security requirements for the scenario, which are defined and supported by the DESEOS Security Architecture [3], particularly, requirements for authentication and authorization. In the following we summarize the most relevant security requirements.

- *Confidentiality and privacy* of multimedia data, such as audio and video, when transmitted across local and Internet network.
- *Authentication* of network entities such as school information servers, authentication of pupils when given access to the school multimedia data.
- *Accountability* of pupils when accessing media data of the school information system. This requirement is closely related with authentication and confidentiality requirements.
- *Access control* to school-related multimedia resources accessed by pupils. A proper certificate-based controlled access mechanism ensuring decentralized trust establishment.

3.3 Software architecture

Figure 3 shows the software architecture of LIFE, and how the LIFE application realizes the client-server model of the reference architecture. The software architecture takes advantage of DESEOS Core [3] to connect the realms of hospital and

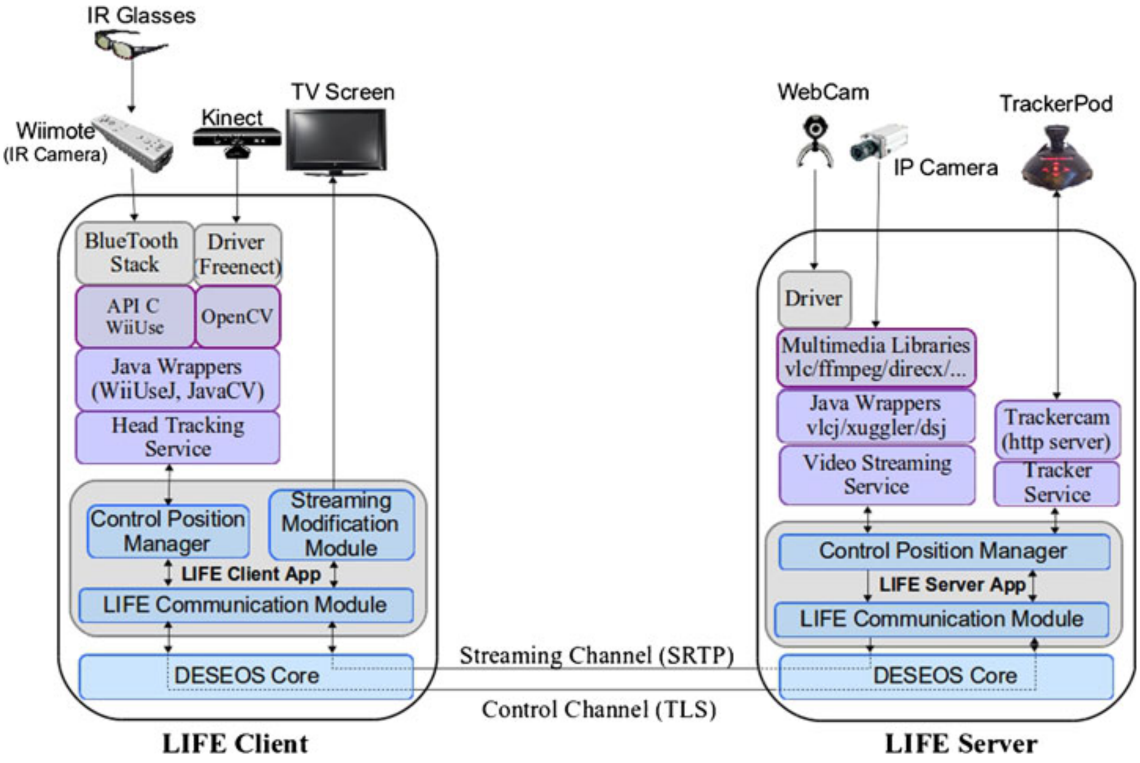


Fig. 3 LIFE software architecture

school. Additionally, the architecture implements several services, not part of the conceptual model, to carry out the immersive view effect, such as Head Tracking, Video Streaming and Tracker Services.

DESEOS Core is in charge of establishing appropriate communication channels with suitable security protocols. This provides a flexible approach to adopt different protocol solutions in the future to enable secure communication without affecting the LIFE architecture. Each realm involves the use of several devices, which can be replaced by others with similar functionality. To achieve that particular feature, both DESEOS Core and LIFE application make use of OSGI Framework [21], since this provides an easier way to work with services. Thus, we highlight the feature that allows to connect LIFE with heterogeneous devices in a secure way.

On the server side, the LIFE Server App makes use of two different services, Video Streaming Service and Tracker Service. The former is used to provide a video streaming which will be transmitted to client applications using the SRTP protocol. We notice that there are several implementations of the Video Streaming Service component depending on the actual video capture device (webcam or IP camera) and the multimedia libraries used (Xuggler,[3] vlcj,[4] dsj,[5] etc). The latter is in charge of moving a camera device to a specific position.

[3] http://www.xuggle.com

[4] http://caprica.github.com/vlcj

[5] http://www.humatic.de/htools/dsj.htm

The PTZ devices have rotation properties such as turn right/left (Pan) and turn up/down (Tilt). A special property is the zoom capability, provided by some PTZ devices if a direct connection to the camera device exists. In our case, the zoom is done by the software. Currently, it is implemented by means of the TrackerCam component which has an associated HTTP server in charge of accepting, processing and performing rolling of the physical device.

On the client side, the LIFE Client App makes use of the Head Tracking Service to get head position data according to some anchor point. Two different implementations have been developed to address this idea, one based on the WiiUseJ[6] library, which is used to access the Wiimote (remote control of the Wii system). Another one based on the OpenKinect[7] library that allows to access the Kinect (3D sensor device of the XBox 360).

The WiiMote approach is implemented by using a infrared wiimote sensor, which locates user position by collecting data from user infrared glasses. The WiiUseJ library functionality is used to process input information from the wiimote sensor. Another implementation is based on the Kinect functionality, which is more complex since OpenKinect library still does not provide accurate methods to get a user's head position. Then we make use of openCV to analyse and recognise Kinect data. This analysis is essentially a face recognition based on identifying the biggest face in the frame as a reference point. The head position coordinates are computed according to the coordinates x and y from the position of the face in the frame, while the z coordinate is obtained by using the depth sensor of the Kinect device.

3.4 Immersive view realization

Almost all the process to achieve the immersive view is carried out on the client side. However, the server is in charge of opening a control socket on a predefined port to wait for a client connection request. When the connection is established some control parameters are set and exchanged, the Control Channel is started and the Multimedia Streaming Channel for streaming communications is initialized.

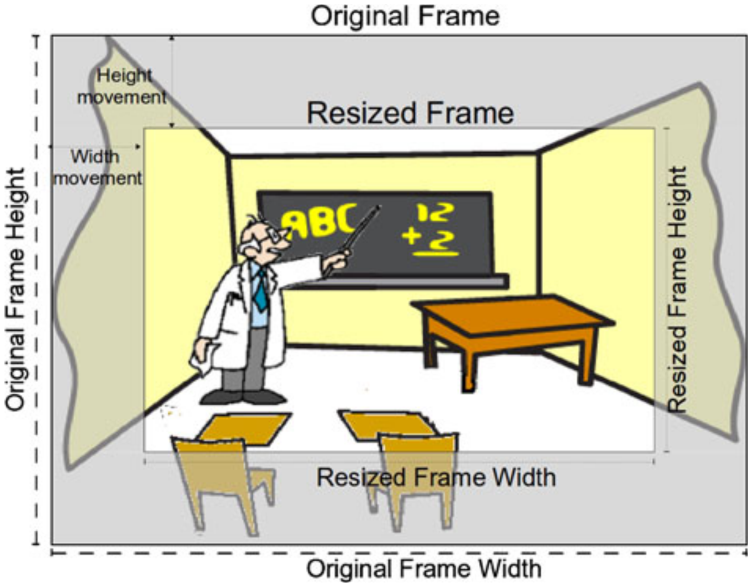
On the client side, the video streaming is received, decoded and sent to LIFE Client App component. This component renders input frames depending on current user's head position to simulate the immersive view. Figure 4 shows the resized frame effect compared to the original frame received. The LIFE Client App computes the values of height and width according to the user position obtained from the Head Tracking Service. Figure 5 shows the tilt-pan calculation in relation to the user's previous position and the user's current position. The equations in the figure show the functions used to calculate *pan*, *tilt* and *zpan* values.

After a testing procedure, we have deduced that in order to have an accurate system it is necessary to get the user head position data as input, at least as often as the video streaming frequency (fps). This requirement is necessary to guarantee smooth movements of the immersive view. The computational overload of the Head Tracking Service (as Kinect solution does by using JavaCV) made us to compute

[6] http://code.google.com/p/wiiusej

[7] http://openkinect.org/wiki/MainPage

Fig. 4 LIFE resized frame effect

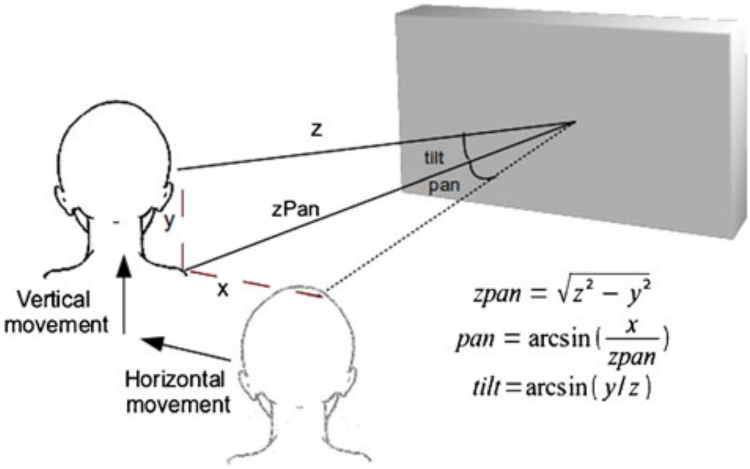


head position coordinates by means of interpolation to reach a smoother movement and therefore a more realistic user immersion. The number of interpolation items can vary depending on a head position coordinate rate parameter, in such a way that the optimal settings for the immersive view require a good calibration of the head position coordinate rate parameter.

Additionally, our implementation can be adapted dynamically to some context conditions to reach an optimal efficiency. In this line, for those cases of a limited bandwidth, video streaming parameters can be configured according to these conditions. Thus, frame rate parameter can be set as lower as possible. Consequently, the smoothness is achieved by the internal mechanism implemented using the same frame several times and just changing the head position coordinate.

An error correction mechanism has been implemented to discard error flow positioning returned by devices (senseless data). Essentially, the mechanism behaves as if two consecutive coordinates differ more than a reference predefined control value, the last coordinate is discarded, but if this happens more than N times (the value of N can be configured) in a row, we get the last coordinate as a valid

Fig. 5 LIFE tilt-pan calculation



one and the error correction mechanism is restarted. In this way, instantaneous head movements are discarded unless the head position remains a longer time in one zone range.

When the resized frame has to be moved beyond original video resolution, (frame boundaries in Fig. 4) Life Client App sends a move command through the Control Channel to position the video capture device to a new region. We have defined a set of predefined areas (spaces) where a PTZ-type device should move to. The granularity of these areas depends on the accuracy of movements of an current PTZ device in order to avoid unnecessary noise (shake) movements. When the resized frame has to move to a new position beyond the original frame, the frame controller computes the next area where the current PTZ device should move and use the control channel to set the video capturing device to cover the new position of the resized frame.

3.5 Communications security

An important aspect of LIFE practical adoption is the addressing of the security of LIFE communications. We adopted X.509 [16] certificates to encode the necessary information about pupils and schools. We define a set of subject-specific attributes that allows us to express information of an entity in the LIFE scenario necessary for authentication, access control and communication establishment.

Figure 6 shows security communications of LIFE application scenario. When a secure and trusted channel with the LIFE server is needed the Control Channel is established. The communication on this channel (LIFE client and LIFE server)

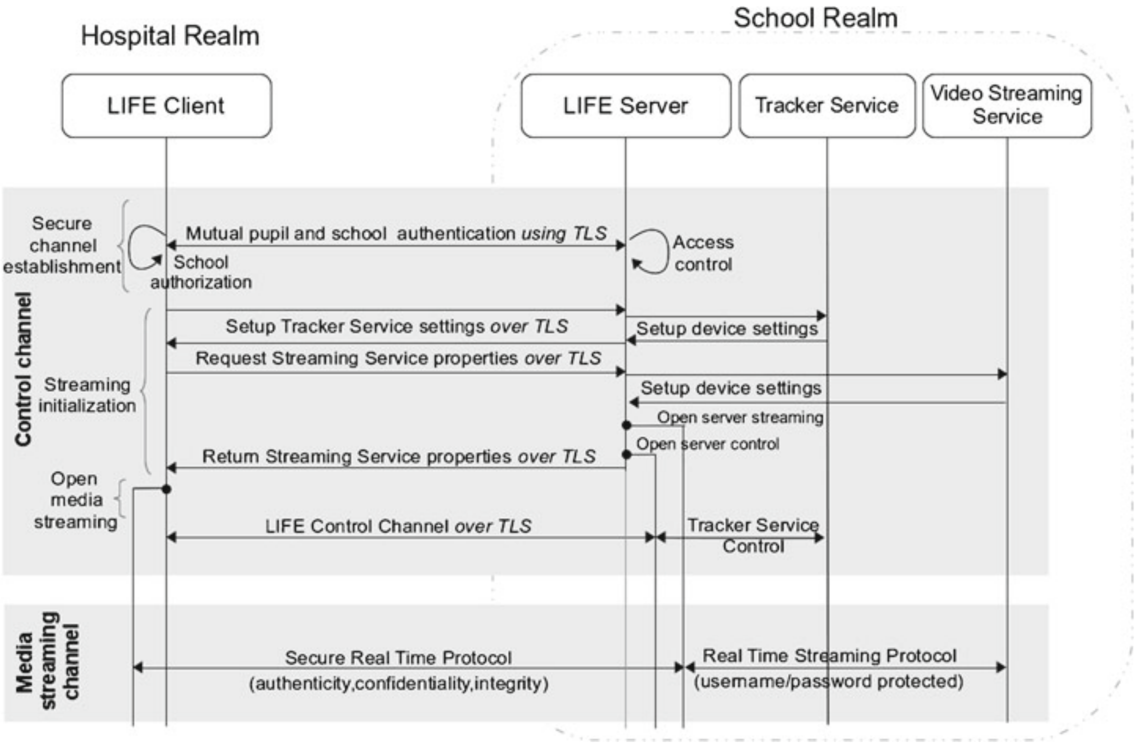

Hospital Realm
School Realm
LIFE Client
LIFE Server
Tracker Service
Video Streaming Service
Control channel
Secure channel establishment
Mutual pupil and school authentication using TLS
School authorization
Access control
Streaming initialization
Setup Tracker Service settings over TLS
Setup device settings
Request Streaming Service properties over TLS
Setup device settings
Open server streaming
Open server control
Return Streaming Service properties over TLS
Open media streaming
LIFE Control Channel over TLS
Tracker Service Control
Media streaming channel
Secure Real Time Protocol
(authenticity,confidentiality,integrity)
Real Time Streaming Protocol
(username/password protected)

Fig. 6 LIFE communications security

entails the use of pupil and school certificates using the TLS protocol.[8] An access control process has been implemented (after TLS handshake), such that on the LIFE server side is verified whether the pupil data in the certificate states the correct school, year of study and class id. We evaluate the year of study and class id to ensure that the current pupil will have access to the correct multimedia streaming. The protocol considers the case of parallel connections of more that one pupil to the same school. Then in the case of more than one pupil of the same year and class requesting access to the streaming channel, the LIFE server configures the multimedia streaming to passive mode, which consists on positioning the camera in the initial calibration state and disabling the control functionality on the Tracker Service. In that way, none of the pupils has control on the camera and the immersive view on the LIFE client application uses only the resized frame mechanism (of software-simulated immersion).

A school authorisation process on the side of the LIFE client has been implemented, which ensures that the school certificate is not only valid and trusted (a valid DESEOS school entity) but also authorised, i.e. if the school (by name, locality and country) matches to the school in the pupil certificate. In that way, confidentiality and privacy of pupil's communications with the correct school are enforced.

Once a secure and trusted channel is established the next step is the initialisation of the streaming, which is performed over the secure channel established (over TLS). It has two steps: Tracker Service initialisation and Video Streaming Service initialisation. The first step determines if a Tracker Service is enabled, while the second step determines the Video Streaming Service properties of current camera settings and some security parameters for the media streaming channel. When both of them are initialised, the LIFE server opens a control channel and a media streaming channel, and then returns the media streaming properties object to the LIFE client over the control channel. In turn, the LIFE client application opens a media streaming channel with the indicated settings and the streaming protocol. We remark that the control channel, running over TLS, is used only for immersive view control commands (not for media streaming) so that the induced overhead of TLS does not affect the immersive view effect.

We have adopted the usage of SRTP for protecting media streaming channel with confidentiality, authenticity and integrity. The LIFE control channel remains over the TLS channel already established. The LIFE control channel is used for exchanging different commands, for instance with the Tracker Service upon movements by the pupil, while the streaming channel is returning the media of the camera.

As it was mentioned in Section 4, the SRTP has been specifically designed to provide core security properties with a strong level of security by using well known security building blocks and algorithms. At the same time, it provides efficient security processes with a minimal additional cost of data transmission, which is an important aspect for live media streaming. There is a master key and a master salt cryptographic element as part of the SRTP configuration, which the LIFE server sends to the LIFE client along with the media streaming properties during initialisation over the secure control channel.

[8] http://tools.ietf.org/html/rfc5246

Table 1 Hardware specification

	LIFE client	LIFE server
CPU	Pentium dual-core E5300 2.6 GHz	Intel core duo T2500 2 GHz
Memory	4 GB DDR2 800 MHz	1 GB DDR2 533 MHz
Graphics	NVidia GeForce 9400 GT	NVidia GeForce Go 7400 128 MB
OS	Ubuntu 10.04 LTS	Windows XP

The LIFE server generates a new master password and master salt for any LIFE client application authorised to access media streaming of the school. This means that if more than one pupil access to the streaming of the school they will be using different master keys and salt, and will respectively derive different session keys protecting the streaming data. The LIFE sever keeps each master key only while the session is open(e.g., the current configuration is per 8 h).

3.6 Streaming performance evaluation

LIFE evaluation has been done with a hardware specification shown in Table 1 and with a GrandStream GXV 3601 IP-Camera. All software used in LIFE is open source, allowing us to set in code points of measurement to obtain necessary data for our evaluation. The main software package used is VLC 1.2.0, which makes use of mpeg (libavcodec 53.2.0, libavformat 53.0.3), x264 codec and live555 Streaming Media library.

In order to show the real frame processing time, we have excluded network delays from the measurements. Therefore, all evaluations have been done in a LAN ensuring enough bandwidth for media streaming. Another consideration to take into account is that the encryption and decryption operations consume the same computational time (are the same operation) due to the SRTP underlying cryptographic mechanisms. Therefore, we measured the security overhead time on the LIFE-Server and multiplied it by two to get the total security process time.

In order to see how the security process affects video streaming, we have extended the computation processes to different fps (frame per seconds) configurations. Figure 7 shows the performance details of LIFE streaming for both, secure and non-secure versions for two video resolutions. Each of the graphics represents the video processing time in ms on y axis over different fps on x axis. The non-secure

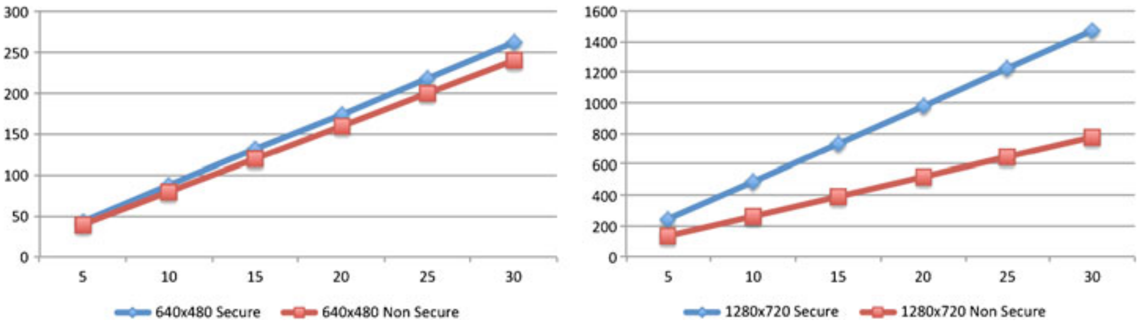


Fig. 7 LIFE streaming performance evaluation for 640 × 480 and 1280 × 720 resolution

Table 2 LIFE streaming performance per frame resolution and frames per second (secure/non-secure) in ms

Res\fps	5	10	15	20	25	30
640 × 480	43,6/40	87,2/80	130,8/120	174,4/160	218,1/200	261,7/240
800 × 480	88,9/60	177,9/120	266,8/180	355,8/240	444,8/300	533,7/360
800 × 592	103,9/75	207,9/150	311,8/225	415,8/300	519,8/375	623,7/450
1024 × 768	208,6/120	417,3/240	626,0/360	834,7/480	1043,4/600	1252,1/720
1280 × 720	245,8/130	491,6/260	737,5/390	983,3/520	1229,2/650	1475/780
1280 × 960	280,8/165	561,6/330	842,5/495	1123,3/660	–	–

video streaming (red line) is compared to the secure one (blue line) to show how the security modules inclusion affects the efficiency in terms of time consumption. More details of the security performance of different video resolutions can be found in Table 2.

An analysis of the main result, in the first case (640 × 480), shows that the difference between the secure and non-secure streaming is almost negligible even in the case of 30 fps (21 ms). However, we are interested in inspecting the security overhead for higher video resolutions. We show in Fig. 7 the media streaming performance for the highest resolution. In the case of 1280 × 720 the upper bound of feasible secure streaming is 20 fps, taking into account that network delay is not included in the evaluations. Respectively, in the case of 1280 × 960 the upper bound is reduced to 15 fps.

A reference point for the above conclusions is the one-second threshold of process time in order to have normal video behaviour. We have considered that all measurements above that threshold are non-viable cases since the processing time exceeds more than a second (impossibility to show all frames per second).

3.7 Network settings impact

We have considered the end-point processing time including security and its impact on LIFE media streaming performance. However, there are also some network-related aspects such as bandwidth, packet loss and packet delay variation that impact on the LIFE streaming performance. Generally, any live media streaming is vulnerable and sensitive to these aspects. On the other side, given the way SRTP works, the protection of media streaming data by SRTP is as sensitive to network settings as the RTP protocol is without security. In that context, any network measures against packet loss or packet delay variation that apply to the RTP media streaming could also be applied to the SRTP streaming, since the RTP payload is secured and the SRTP packets structure remains as processable as the RTP packet (refer to [5] for details).

The bandwidth aspect mostly impacts on the live streaming performance. In case of expected low bandwidth on the side of LIFE clients (hospitals), the calibration module of the LIFE application will allow the administrator entity to lower either the frames per second or the video resolution, or both of them, in order to optimize the streaming data to the bandwidth. In the concluding remarks below we provide some optimal settings of LIFE application taking into account some possible bandwidth restrictions.

4 Related work

There are several different areas of related work: virtual window approaches, real-time protocols for streaming media, security of media streaming solutions, and remote media space monitoring.

4.1 Virtual windows

There is a certain number of works that couple the movements of the observer's head to the shifts in the image [1, 7] but none of these systems took into account the idea of the fixation point as the link between the observer movements and the image shift. Overbeeke and Stratmann [23] proposed the first method for three dimensional image presentation based on a fixation point as linker. Based on this method a Virtual Window system was proposed [11]. This system uses head movements in the viewer location to control camera shifts in a remote location. As a result, viewers have the impression of being in front of a window allowing exploration of remote scenes rather than a flat screen showing moving pictures. One of the most relevant goals of this approach is to present an immersive sensation to the viewer, but due to some drawbacks it was not reached. Among these, we highlight the fact that the techniques used were not sufficiently efficient to achieve a fluent video streaming and an accurate head tracking.

The Virtual Window effect is achieved by means of video capture device movements, but not considering frame resize approach due to a limited video resolution. In our settings, we improved the Virtual Window effect by exploiting the resized frame approach over high definition video quality.

The proposal by Chung Lee [17] focuses on head tracking. It uses the infrared camera (Wiimote) and a head mounted sensor bar (two IR LEDs) to accurately track the location of user head modifying the screen view according to the anchor point of the user. However, this work is limited to the Wii remote device. Rational Craft has commercialized a product based on the Chung Lee approach that plays locally recorded videos.[9] These two approaches are based on the use of 3D models and recorded videos, respectively, instead of real video streaming as required in our use case scenario, which introduces an additional layer of complexity.

4.2 Real-time media streaming solutions

Regarding the design and implementation of real-time protocols for streaming several approaches have been proposed. The RTP (Real Time Protocol) [25] is an RFC standard and one of the first solutions for real-time communications in 1996. Simultaneously, the most relevant companies developed their proprietary solutions as Microsoft Netshow or MMS, ex-Macromedia RTMP, etc [6].

A tailored version of this protocol for streaming is the RTSP (Real Time Streaming Protocol) [26] appeared in 1998, which included specific interfaces for stream control, playing, stopping, etc. Many different tailored protocols derived from this

[9] http://www.rationalcraft.com/Winscape

have been developed for particular cases such as the SRTP [5] for security purposes, SIP [24] for session and distribution goals or the WebRTC[10] for a browser-to-browser streaming.

Nowadays, there are several commercial approaches for real-time P2P streaming such as Octoshape[11] and PPLive.[12] A comprehensive survey of P2P media streaming systems can be found in [29]. Octoshape has been used to broadcast live streaming and help CNN serve a peak of more than a million simultaneous viewers. It provides several delivery technologies such as loss-resilient transport, adaptive bit rate, adaptive path optimization and adaptive proximity delivery. The Octoshape solution splits the original stream into a number K of smaller equal-sized data streams but a number $N > K$ of unique data streams are actually constructed. In such a way, a peer receiving any K of the N available data streams is able to play the original stream.

PPLive, one of the most popular P2P streaming software in China, consists of several parts: (i) Video streaming server: providing the source of video content; (ii) Peers; (iii) Directory server: automatically registers user information to and cancels user information from PPLive clients; and (iv) Tracker server: records information of all users watching the same content; When the PPLive client requests some content, the tracker server checks if there are other peers owning the content and sends the information to the client. PPLive uses two major communication protocols: Registration and Peer Discovery protocol, and P2P Chunk Distribution protocol.

Architecturally, P2P streaming solutions have different goals compared to our client-server model of immersive view. These, however, can be used to provide ground for extending the reference architecture to enable distribution of media streaming data over P2P topology, in cases multiple pupils wish to connect to a remote space of the school. As we have discussed in Section 3.5, a specific solution providing immersive view to multiple peers (pupils) is to enable passive remote space control with only software-simulated client-side immersive view.

4.3 Security of real-time media streaming

The LIFE application integrates the Secure Real-time Transport Protocol (SRTP) for protecting live media data in streaming. The SRTP, a profile of the RTP, aims at providing confidentiality, message authentication, and replay protection to the RTP-based media streaming. Its main goal is to enable strong cryptographic operations and, at the same time, high throughput with low packet extension (minimal additional cost of data transmission). If using the default encryption settings of SRTP *the RTP payload and the SRTP payload have exactly the same size*.

Essentially, the SRTP implementation is a “bump in the stack” between the RTP application and the transport layer. The SRTP intercepts RTP packets down the stack, performs secure operations on the packet, and forwards an equivalent SRTP

[10] http://www.webrtc.org

[11] http://www.octoshape.com

[12] http://www.pplive.com

packet to the sending node. On the receiver side, the SRTP intercepts SRTP packets, performs secure operations on the SRTP packet, and passes an equivalent RTP packet up in the stack (to the RTP application).

The underlying cryptographic blocks are an additive stream cipher for encryption and a keyed-hash function for message authentication and integrity. The default master key length is 128 bits and 112 bits for the master salt. The encryption and decryption process of RTP payload have the same computational operations and, consequently, the same computing cost.

Encryption:	AES-CM with 128 bits session key and 112 bits session salt.
Authentication:	HMAC-SHA1 with 128 bits session key and 112 bits session salt.

Widener et al. [28] propose an approach of differential data protection where different portions of the media streaming could have different protection policies enforced during streaming. Authors identify three types of policies on video streaming: general protection policy regardless of any image streams, policy governing access to image streams, and policy governing access to filters for a particular stream. In order to request an image stream an entity needs a credential with assigned rights for that. A credential contains a collection of general access rights and a collection of rights specific to object. Access rights define peer accessibility to portions of streaming data and specify what filters have to be used to process media data after acquiring the streaming.

Liao et al. [19] follow the approach [28] applied to the domain of secure media streaming over peer-to-peer networks. Upon initial joining of peer in the streaming system, the peer contacts a pre-defined authentication server to obtain a credential with similar structure and usage as in [28].

The works of Widener et al. and Liao et al. focus on streaming protection by means of certificates and policies with local or middleware monitoring and enforcement of the expressed rights. In their approaches the stream data is left unprotected during network transmission and easy to intercept/modify by other (malicious) peers. Even more, recent studies on security and privacy issues in peer-to-peer streaming systems [12] show that commercial streaming solutions do not perform encryption protection on data during transmission, which makes the overall system loose not only confidentiality but also authenticity and privacy.

Our approach provides complementary solution to [19, 28] in the way that we adopt the usage of certificates to enable an access control process between two entities establishing requested media stream, and exchange of cryptographic keys used to secure streaming data integrity and confidentiality during transmission. Thus, any malicious peer intercepting (joining) the streaming will not gain access to the streaming data.

4.4 Remote media space monitoring

The third area of research related with our approach is the control of remote media space for video monitoring. Most of the improvements in this area have focused on the inclusion of dedicated hardware in the camera device to compute the monitoring itself, the most evident case is the widespread use of the PTZ cameras dedicated to video surveillance [9, 18]. Many efforts have been spent in the improvements of object monitoring (persons, animals, etc.) in a limited area monitored by a certain number of cameras.

Instead of shifting the camera following the movements of a recorded object, our approach follows a different perspective by deducing camera movements according to user's motion (e.g., user's head location/position).

5 Conclusions

We have presented a reference architecture conceptualizing the immersive view effect by considering various heterogeneous devices for observer's position tracking and an enriched movement control over remote multimedia streaming sources. We have also presented a proof-of-concept implementation, called LIFE, which enables an immersive view effect of school activities during children hospitalization. The goal of LIFE is to contribute to a vision of future AAL applications for "Alleviating Children Stress During Hospitalisation" [2]. Functional and security aspects of LIFE have been also presented along with implementation details of how these aspects have been achieved.

Given the targeted group of users - kids of age ranging from 8 to 16 years old, an important concluding aspect of LIFE presentation is to identify a set of settings optimised to achieve a balance between performance and quality of the immersive view effect. There are two main parameters to take into account: total time of processing secure video streaming (of given frames per second), and video resolution enabling us to maximise the immersive view effect. The work in [4] reports several comparative details on expected watchability percentage over reduced frame rates of video processing. Depending on several parameters, the authors come to the conclusion that 10 fps results in 80 % expected watchability, while 15 fps or higher results in 90 % or higher. There are several comparison results between 5 to 10 fps and 10 to 15 fps with an interesting conclusion that in some conditions for 5 fps 80 % watchability can be achieved.

Given the expected pupils' age, we concluded that 10 fps is the lower bound to avoid any emotional discomfort of pupils using LIFE technology for long periods. According to Table 2, secure media streaming is feasible for all video resolutions under 10 and 15 fps, while increasing above 15 fps the feasible resolution decreases. Using 30 fps secure media streaming is feasible up to 800 × 592 (given the granularity of resolutions tested).

The video resolution settings allow us to increase the quality of video streaming and maximise the immersive effect of resized frames on pupil's movements. The higher the resolution is, the more space is available to move the resized window without using the trackerpod for changing the camera angle. The cons of this setting are the need to transport much larger amount of data, which implies a higher bandwidth and probably more network delay.

We conclude that the optimal settings of LIFE video streaming is 1280 × 720 pixels of 10 fps. The results show that even without a high bandwidth connection LIFE application is still able to provide good quality of immersive view experience for the defined optimal settings taking into account potential network delay. We argue that the higher resolution of 1280 × 960 only gives more vertical space for the resized frame, which is the less important axis (of use) for pupils.

6 Future work

Future work includes the development of a calibration module for LIFE able to adjust video resolution and fps parameters to optimal values for a given user taking into account quality of network connection and perception aspects of the user. From a security point of view, the reference architecture can be enriched by supporting DESEOS security pattern framework [3] in order to flexibly address security solutions which may be adopted in different application domains. Another future work field is to address an integration of the LIFE application as an AAL service, which can facilitate several communication issues by means of the underlying platform's facility. Nowadays, there are several projects offering a platform for AAL services, such as [14, 27].

Finally, a future work will focus on refining and formalizing the commands format and communication protocol of LIFE to support enriched control of remote media spaces with heterogeneous image sources.

Acknowledgements This work is supported by the project **DESEOS** (TIC-4257) **D**ispositivos **E**lectrónicos **S**eguros para la **E**ducación, **O**cio y **S**ocialización (meaning "secure electronic devices for education, entertainment and socialization") funded by the government of Andalucía.

References

1. Amos B, Wang M (1978) Stereo television viewing for remote handling in hostile environments. In: Proc. conf. remote syst. technol., United States
2. Antón P, Maña A, Muñoz A, Koshutanski H (2011) Live interactive frame technology alleviating children stress and isolation during hospitalization. In: 3rd international workshop on ambient assisted living (IWAAL 2011), Málaga, Spain, pp 92–100
3. Antón P, Muñoz A, Maña A, Koshutanski H (2012) Security-enhanced ambient assisted living supporting school activities during hospitalisation. J Ambient Intell Human Comput 3(3):177–192
4. Apteker RT, Fisher JA, Kisimov VS, Neishlos H (1995) Video acceptability and frame rate. IEEE Multimed 2(3):32–40
5. Baugher M, McGrew D, Naslund M, Carrara E, Norrman K (2004) The secure real-time transport protocol (SRTP). RFC 3711. http://www.ietf.org/rfc/rfc3711.txt
6. Chen W, Shih C-C, Hwang L-J (2010) The development and applications of the remote real-time video surveillance system. Tamkang J Sci Eng 13(2):215–225
7. Clarke MM (1982) Remote systems: some human factors issues in teleoperator and robot development: an interactive session. Proceedings of the Human Factors and Ergonomics Society Annual Meeting 26(9):763–765
8. Costa R, Carneiro D, Novais P, Lima L, Machado J, Marques A, Neves J (2009) Ambient assisted living. In: 3rd symposium of ubiquitous computing and ambient intelligence. Advances in soft computing, vol 51, pp 86–94
9. Dinh T, Yu Q, Medioni G (2009) Real time tracking using an active pan-tilt-zoom network camera. In: 2009 IEEE/RSJ international conference on intelligent robots and systems. IEEE Press, Piscataway, pp 3786–3793
10. Dowden DC, Gitlin RD, Martin RL (2002) Next-generation networks. Bell Labs Tech J 3(4):3–14
11. Gaver WW, Smets G, Overbeeke K (1995) A virtual window on media space. In: CHI, pp 257–264
12. Gheorghe G, Lo Cigno R, Montresor A (2011) Security and privacy issues in P2P streaming systems: a survey. Peer-to-Peer Netw Appl 4(2):75–91
13. Giles J (2010) Inside the race to hack the kinect. New Scientist 208(2789):22–23

14. Hanke S, Mayer C, Hoeftberger O, Boos H, Wichert R, Tazari M-R, Wolf P, Furfari F (2011) universAAL an open and consolidated AAL platform. In: Ambient assisted living. Springer, Berlin Heidelberg, pp 127–140
15. Ingebretsen M (2010) In the news. IEEE Intell Syst 25(4):4–8
16. Kuhn DR, Polk WT, Hu VC, Chang SJ (2001) Introduction to public key technology and the federal PKI infrastructure. Technical report
17. Lee JC (2008) Hacking the Nintendo Wii remote. IEEE Pervas Comput 7(3):39–45
18. Li Y, Bhanu B, Lin W (2010) Auction protocol for camera active control. In: 2010 IEEE International conference on image processing. IEEE Press, Piscataway, pp 4325–4328
19. Liao R, Yu S, Yu J (2007) SecureCast: a secure media streaming scheme over peer-to-peer networks. In: Workshop on intelligent information technology application (IITA 2007). IEEE Press, Piscataway, pp 95–98
20. Lockin M (2005) The redefinition of failure to thrive from a case study perspective. Pediatr Nurs 31(6):474–479
21. Marples D, Kriens P (2001) The open services gateway initiative: an introductory overview. IEEE Commun Mag 39(12):110–114
22. Muñoz Hoyos A (1996) Influencia de la institucionalización sobre el crecimiento, desarrollo y comportamiento en el niño. Part of the course: principales problemas sociales en la infancia. Educación y cuidados. Escuela Universitaria de Ciencias de la Salud, Granada
23. Overbeeke CJ, Stratmann MH (1988) Space through movement. A method for three-dimensional image presentation. PhD thesis, Technische Universiteit Delft
24. Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Sparks R, Handley M, Schooler E (2002) SIP: session initiation protocol. Technical report, RFC 3261
25. Schulzrinne H, Casner S, Frederick R, Jacobson V (2003) RTP: a transport protocol for real-time applications. RFC 3550
26. Schulzrinne H, Rao A, Lanphier R (1998) Real time streaming protocol (RTSP). RFC 2326. http://www.ietf.org/rfc/rfc2326.txt
27. Tazari M-R, Furfari F, Lázaro Ramos J-P, Ferro E (2010) The PERSONA service platform for AAL spaces. In: Handbook of ambient intelligence and smart environments. Springer, US, Boston, MA, pp 1171–1199
28. Widener P, Schwan K, Bustamante FE (2003) Differential data protection for dynamic distributed applications. In: Proceedings 19th annual computer security applications conference, 2003, pp 396–405
29. Yingjie G, Piccolo F, Duan S, Zhang Y, Zong N (2013) Survey of P2P streaming applications. Technical report. http://tools.ietf.org/html/draft-ietf-ppsp-survey-04