UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

CONTRIBUTION TO SIGNALLING OF 3D VIDEO STREAMS IN COMMUNICATION SYSTEMS USING THE SESSION INITIATION PROTOCOL

TESIS DOCTORAL

PEDRO CAPELASTEGUI DE LA CONCHA

2012

Departamento de Ingeniería de Sistemas Telemáticos

Escuela Técnica Superior de Ingenieros de Telecomunicación

Universidad Politécnica de Madrid

CONTRIBUTION TO SIGNALLING OF 3D VIDEO STREAMS IN COMMUNICATION SYSTEMS USING THE SESSION INITIATION PROTOCOL

TESIS DOCTORAL

Autor
Pedro Capelastegui de la Concha
Ingeniero de Telecomunicación

Director
Francisco González Vidal
Doctor Ingeniero de Telecomunicación

Madrid 2012

Título: contribution to signalling of 3d video streams in communication systems using the session initiation protocol

Autor: Pedro Capelastegui de la Concha

Director: Francisco González Vidal

Tribunal Calificador

Presidente

Vocales

Secretario

Realizado el acto de defensa y lectura de tesis en Madrid el día

Calificación:

El presidente Los Vocales

El secretario

Abstract

3D video technology has been on the rise in the last years, with abundant research advances accompanied by a widespread adoption by the film industry and an increasing importance in consumer electronics. On a related note, there is the concept of multiview video, which encompasses 3D video, and can be defined as a video stream composed of two or more views. Multiview video enables advanced video features such as stereoscopic video, free viewpoint video, improved eye contact through use of virtual views, or shared virtual environments.

The purpose of this thesis is to address a major obstacle towards the use of multiview video in communication systems – the lack of support for this technology from existing signalling protocols, which makes it impossible to configure a multiview video session through standard means. Thus, our main objective is to extend the Session Initiation Protocol (SIP) to support the negotiation of multimedia sessions with multiview video streams.

Our work can be summarized in three major contributions. In the first place, we have defined a signalling extension for the configuration of SIP sessions with 3D video. This extension changes the Session Description Protocol (SDP) to introduce a new media-level attribute and a new type of decoding dependency, which help describe the 3D video formats that can be used in a session, and the relationship between video streams composing a 3D video stream.

The second contribution consists in a SIP extension to handle the signalling of video conferences with multiview video streams. Two new SIP event packages are defined to describe the capabilities and topology of conferencing terminals, on the one hand, and the spatial configuration and stream mapping of a conference, on the other. A way to integrate the exchange of this information into the initiation of a SIP conference is also described.

For the third and final contribution, we introduce the concept of conference virtual space, as a coordinate system where all relevant objects in a conference (such as capture devices, displays, and users) are included. We explain how the virtual space relates to conference features such as eye contact, video scale and spatial faithfulness, and provide guidance on how to determine the features of a conference through virtual space analysis, and on the generation of virtual spaces as part of conference configuration.

Keywords

3D video, multiview video, multimedia session, SIP, SDP, signalling ,telepresence

Resumen

Las tecnologías de vídeo en 3D han estado al alza en los últimos años, con abundantes avances en investigación unidos a una adopción generalizada por parte de la industria del cine, y una importancia creciente en la electrónica de consumo. Relacionado con esto, está el concepto de vídeo multivista, que abarca el vídeo 3D, y puede definirse como un flujo de vídeo compuesto de dos o más vistas. El vídeo multivista permite prestaciones avanzadas de vídeo, como el vídeo estereoscópico, el “free viewpoint video”, contacto visual mejorado mediante vistas virtuales, o entornos virtuales compartidos.

El propósito de esta tesis es salvar un obstáculo considerable de cara al uso de vídeo multivista en sistemas de comunicación: la falta de soporte para esta tecnología por parte de los protocolos de señalización existentes, que hace imposible configurar una sesión con vídeo multivista mediante mecanismos estándar. Así pues, nuestro principal objetivo es la extensión del Protocolo de Inicio de Sesión (SIP) para soportar la negociación de sesiones multimedia con flujos de vídeo multivista.

Nuestro trabajo se puede resumir en tres contribuciones principales. En primer lugar, hemos definido una extensión de señalización para configurar sesiones SIP con vídeo 3D. Esta extensión modifica el Protocolo de Descripción de Sesión (SDP) para introducir un nuevo atributo de nivel de medios, y un nuevo tipo de dependencia de descodificación, que contribuyen a describir los formatos de vídeo 3D que pueden emplearse en una sesión, así como la relación entre los flujos de vídeo que componen un flujo de vídeo 3D.

La segunda contribución consiste en una extensión a SIP para manejar la señalización de videoconferencias con flujos de vídeo multivista. Se definen dos nuevos paquetes de eventos SIP para describir las capacidades y topología de los terminales de conferencia, por un lado, y la configuración espacial y mapeo de flujos de una conferencia, por el otro. También se describe un mecanismo para integrar el intercambio de esta información en el proceso de inicio de una conferencia SIP.

Como tercera y última contribución, introducimos el concepto de espacio virtual de una conferencia, o un sistema de coordenadas que incluye todos los objetos relevantes de la conferencia (como dispositivos de captura, pantallas, y usuarios). Explicamos cómo el espacio virtual se relaciona con prestaciones de conferencia como el contacto visual, la escala de vídeo y la fidelidad espacial, y proporcionamos reglas para determinar las prestaciones de una conferencia a partir del análisis de su espacio virtual, y para generar espacios virtuales durante la configuración de conferencias.

Palabras Clave

Video 3D , video multivista, sesión multimedia, SIP, SDP, señalización,telepresencia

Agradecimientos

A Paco, por su paciencia infinita, y por su magnífica labor como jefe y director. A mis compañeros del RSTI, por su apoyo durante estos años.

A Marisol y a mis niñas, por sufrir mis ausencias durante la redacción de esta tesis, y darme fuerzas para terminar. A mi familia, que me ha animado en todo momento.

A mis amigos.

Me falta tiempo y espacio para agradecer lo suficiente a todos los que me han acompañado en este viaje. Espero tener ocasión de hacerlo en condiciones en persona, ya sin presiones de los plazos de entrega.

Table of contents

Index of figures

Index of tables

1 Introduction

1.1 Context and Motivation

3D video technologies have experienced a significant growth in the last years. The trend started with the wide adoption of glass-based stereoscopic video in film theatres, and has continued with the market introduction of consumer devices capable of recording and reproducing stereoscopic video, including TV sets, cameras, home video consoles and portable gaming devices.

In the academic world, this interest in 3D video has translated into a series of advances affecting the whole video processing chain: capture, encoding, render, transmission, and display. Moreover, this has not been limited to just stereoscopic video (which can be defined as a video stream, usually composed of two views, which includes depth information of a scene), but also to a group of related technologies called multiview video.

Multiview video generalizes the concept of stereoscopic video, to describe a video stream composed of two or more views, and which may also include a model describing the geometry of the represented scene. The use of additional video views allows a multiview video stream to offer new functionality such as the ability to change the perspective of a rendered scene.

Both stereoscopic video and multiview video are rapidly approaching technological maturity, and are supported by media standards such as Multiview Video Coding [MVC], HDMI [HDMI], or MPEG-C Part 3 [MPEG-C-PART3]. However, there is currently one area where support for multiview video is lacking, which might create a bottleneck in the deployment of communication applications featuring this technology. We are referring to the field of network signalling, and specifically to the lack of compatibility of current signalling protocol standards with 3D video and multiview video streams. This is the problem that this thesis intends to address.

The author first came in contact with multiview video technologies in 2008, after joining CENIT-VISION [CENIT-VISION], a Spanish-funded research project for development of an advanced video conferencing system. The VISION conferencing prototype would include features such as high definition video, stereoscopic video both with and without glasses, free viewpoint video, and shared virtual environments. Our role was the handling of the communication system, based on the Session Initiation Protocol (SIP) [RFC3261].

Between 2008 and 2011, we worked on a SIP network architecture capable of configuring VISION conferences with dozens of video views, stereoscopic streams, and spatial information. This required the definition of new signalling extensions for SIP, since neither the existing standards nor prior literature about similar conferencing systems provided a way to meet the requirements of VISION on the signalling plane. The signalling solution used in VISION, described in [2011-Perez], was later refined and expanded into what would become the body of this thesis.

1.2 Objectives

The main objective of this thesis is to **extend the Session Initiation Protocol (SIP) to support the negotiation of multimedia sessions with multiview video streams**. This will allow SIP sessions to provide new features like stereoscopic video, free viewpoint video, and shared virtual environments, among others.

This objective can be divided into several partial goals:

- Identify the signalling requirements **for incorporating stereoscopic video in a multimedia session using SIP,** and define an extension for SIP that meets these requirements. We do this on chapter 3.
- Identify the signalling requirements for **incorporating multiview video, and features like free viewpoint video and shared virtual environments, in a video conference using SIP**, and define an extension for SIP that meets these requirements. We do this on chapter 4.
- Define a **model to describe the spatial properties of a videoconference** with multiview video. Describe how this model relates to conference features like eye contact and spatial faithfulness, and provide guidelines to configure the space of a conference. We do this on chapters 4 and 5.

1.3 Structure of the document

The document is structured as follows:

- Chapter 1 (the present chapter), where we state the motivation behind our work and the objectives of the thesis
- In Chapter 2, we provide the state of the art. We present definitions for basic concepts related to multiview video, including a detailed list of the main features and configuration parameters that characterize video communication systems. We then suggest three example application scenarios showcasing multiview video technologies. Finally, we provide an overview of the conferencing prototypes featuring multiview video that have been developed in the last few years.
- In Chapter 3, we define a signalling extension to configure 3D video sessions with the Session Initiation Protocol (SIP) [RFC3261]. The extension consists on a new attribute and decoding dependency for the Session Description Protocol (SDP) [RFC4566], describing the format of a 3D video stream and the relationships between individual media streams composing a 3D video stream.
- In Chapter 4, we study the signalling requirements of SIP conferencing sessions with multiview video, and propose a signalling extension that meets these requirements. The extension consists on a SIP event package for user agents to exchange information about their multiview capabilities and topology, another SIP event package for a conference focus to describe to user agents the virtual space and stream map of a conference, and a configuration process that integrates these elements with regular SIP session initiation.
- In Chapter 5, we expand on the concept of session virtual spaces (introduced in chapter 4), showing how the analysis of these spaces can provide information about

session features. We also provide guidelines for generating a session virtual space, as part of the configuration of a MVV session.

- In Chapter 6, we present our conclusions and lines of future work.

2 State of the Art

2.1 Introduction and definitions

The rapid growth experienced by 3D video technologies in recent years has turned the concept of 3D video conferencing, which had usually been relegated to experimental scenarios or high end enterprise environments, into something that can now be realistically considered for domestic application in the short term.

In this chapter, we will analyse how multiview video techniques such as stereoscopy or free viewpoint video can be applied to communication sessions. First, we provide definitions for basic multiview concepts, and overview the architectures of a typical 3D video streaming system and a 2D communication system. Then, we identify the most important features and configuration parameters that can be found on a video communication system, whether it uses conventional 2D video or multiview techniques. We define three communication scenarios showcasing multiview video, and discuss how they relate to these parameters. Finally, we examine previous works in this area, characterizing them according to this framework, comparing them, and looking for trends.

2.1.1 Multiview video, Stereoscopic video, free viewpoint video

When talking about three-dimensional video, there are two main features that can be provided: stereoscopic video and free viewpoint video. Each one has its own set of requirements and implications for a video transmission system, and they can appear separately or complementing each other.

Multiview Video (abbreviated as **MVV**) refers to any video stream that is composed of more than one video view at some point of its processing chain. Both stereoscopic video and free viewpoint video are examples of MVV streams. In this document, we refer to multimedia sessions containing stereoscopic or free viewpoint video as **MVV sessions**. Likewise, we refer to communication systems capable of initiating MVV sessions as **MVV communication systems**.

Stereoscopic Video, also known as “stereo video” or “3D video”, allows an observer to perceive depth in a scene, by displaying different representations of the scene, or views, for each of the observer’s eyes.

Stereo video systems require at least two different video views to work, though some use more. The actual video information to be stored or transmitted can correspond with the displayed views, or it can be codified in a main video view plus a depth map (as defined below). In the latter case, that information is used on the receiving side to generate the views to be displayed.

A **Depth Map** is an auxiliary video stream associated with a video view, containing depth information for that view. Each pixel in the depth map represents a depth value for the corresponding pixel in the original view, usually codified as a shade of grey.

Free viewpoint Video (abbreviated as **FVV**) lets the user choose the point of view from which a scene is displayed. Typically, a FVV system allows for smooth transitions from one angle of vision to the other by rendering virtual views of the scene, which don't correspond to any single captured video stream, but are generated from several of them. Points of view can thus be arbitrarily selected, but only within a certain operating range that depends on camera configuration.

2.1.2 Multiview Video streaming architecture

The processing chain for a multiview video streaming system [2007-Kubota] includes capture of multiview images, representation of the 3D scene, coding, transmission, rendering and display, as shown in Figure 1. Each step provides its own challenges, and is strongly dependent on others. For example, the choice of 3D representation conditions the number and disposition of cameras in the capture system, as well as the type of available codecs.

In this section, we discuss how the use of MVV techniques affects each step of the processing chain. Note that all of these steps, except for the 3D representation, are also present in non-MVV video systems.

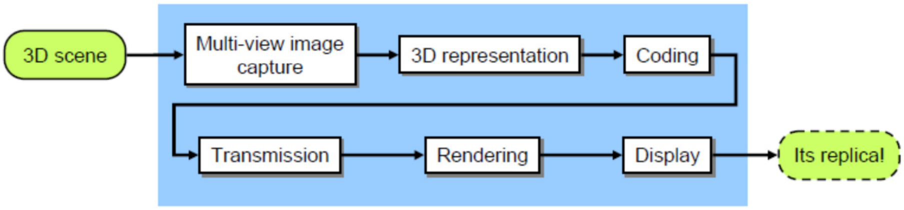

Figure 1 3D Streaming architecture

(Source: [2007-Kubota])

Capture

Multiview video capture for a stereo video system can be performed through a pair of properly positioned conventional cameras, though integrated camera sets also exist that provide stereoscopic views or video plus depth maps. Free viewpoint video applications, on the other hand, require a much higher number of captured views, and involve challenges such as camera calibration and synchronization, and the processing and transmission of large amounts of data.

3D Representation

The method of representation of 3D scenes is a crucial aspect of a multiview video system, since it affects the requirements for video capture and processing. There are two main paradigms for 3D representation, with most methods falling somewhere in between [2000-Kang]. On the one hand, we have geometry-based representation techniques, which use a 3D model to describe the scene geometry, using 3D meshes or other means. The main challenge of a purely geometry-based representation is the cost and complexity of generating the model for a scene, particularly if the scene changes dynamically. On the other hand, there are image-based representation techniques, which use no geometry model, but generate virtual views by interpolation from existing views. Image-based representations have the drawback of requiring a large number of captured views in order to achieve a good rendering quality.

Between the two extremes described above, a common 3D representation mechanism consists in the use of depth maps, or video streams that assign depth values to each pixel of a video view.

For a more in-depth look at 3D representation, [2007-Alatan] provides a survey of existing representation techniques.

Coding

Depending on the choice of 3D representation, a multiview video stream will be composed of one or more video views and, optionally, geometry data streams such as 3D meshes or depth maps. As a consequence, it is important for multiview video applications to have efficient encoding schemes to deal with these large amounts of data.

Regarding video views, the straightforward solution, called simulcast, consists in encoding each view separately. The main drawback of this technique is its lack of efficiency, particularly in scenarios involving many views. In these cases, it is better to use more efficient schemes, like Multiview Video Coding (MVC) [MVC]. This technique , defined as an extension to H.264/AVC [H264] video standard, is based on the simultaneous encoding of multiple video views in order to benefit from inter-view prediction.

As for geometric models, depth maps are typically encoded separately from their base views, as independent video streams. Other types of geometric models, like 3D meshes, are encoded as data streams, requiring dedicated algorithms.

For interested readers, [2007-Smolic] provides a survey of coding algorithms for multiview video.

Transmission

From the point of view of transmission, multiview video presents challenges like bandwidth consumption and backwards compatibility. The bandwidth requirements of a multiview video stream depend on the amount and type of views and geometric models, as well as the choice of coding scheme – the overhead relative to a 2D video stream can range from an extra 10%-30% for certain video plus depth or stereo configurations [2009-Smolic], to multiple times the 2D video bandwidth, for complex streams with many views.

With regards to backwards compatibility, the problem lies in streaming multiview video across intermediate legacy systems that can only handle 2D video – assuming the endpoints at each extreme do support multiview video. For 3D video streams, a way to address this is to use video-plus-depth (also known as MPEG-C Part 3 [MPEG-C-PART3]), which is a standard mechanism that encapsulates a depth map as metadata within a 2D video stream encoded in MPEG-2 or H264/AVC [H264]. Alternatively, it is possible to use frame-packing techniques

[2010-Vetro] to encapsulate a stereo video stream within a 2D video stream using spatial or temporal multiplexing. Frame-packing standards exist for H.264/AVC video [H264] and HDMI [HDMIv1.4a].

Display

Displaying 3D video is only possible with the help of special screens and/or glasses. A variety of 3D display technologies exists, each with their own advantages and drawbacks; we refer the interested reader to [2007-Konrad] for a detailed survey. For the purposes of our study, the most important properties of a 3D display solution are whether or not it requires glasses, the number of displayed views, and the mapping of these views to observing users or observing positions.

The requirement of glasses for certain 3D display solutions imposes restrictions on the kind of applications that can be supported by the system. Glass-based technologies are suitable for video viewing (as demonstrated by their wide adoption in film theaters and, to a lesser degree, TV and video games), but they are inherently limited for two-way communication: most current 3D glasses make it difficult, if not impossible to see the wearer's eyes, which is a major drawback in video conferences. In these scenarios, it is preferable to use glass-free technologies like autostereoscopic displays.

At a minimum, a display needs to show 2 different views in order to convey depth information through stereoscopy. However, configurations that user a higher number of views also exist. In these cases, each displayed view can be observed from a certain range of positions in space. A display has an optimum observing distance, so that a user looking at the display from that distance (or close to it) will perceive a different view on each eye. When enough views are shown this way, users can perceive the perspective of the scene changing when they move – an effect known as **motion parallax**, which further improves depth perception.

2.1.3 2D video sessions

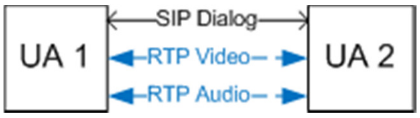

Figure 2 Point to Point 2D video call

Figure 3 2D Video Conference

A **video call** is a multimedia session between two users that includes one or more video streams. Figure 2 shows the general structure of a video call with regular 2D video, and using the Session Initiation Protocol (SIP) [RFC3261] for signalling. On each side of the conversation there is a **User Agent** (UA), which is a logical entity capable of sending and receiving SIP-based signalling messages in order to configure multimedia sessions, as well as exchanging the corresponding media streams. Typical media configurations for 2D video calls comprise one bidirectional video stream for video, and another for audio.

A multimedia session between more than two user agents is called a **conference**. Figure 3 shows a 2D **video conference** that follows the framework for tightly coupled conferencing defined in [RFC4353]. This scenario introduces two new logical functions that are absent from point to point video calls: the focus, and the mixer. The **focus** is a user agent that maintains a signalling relationship with each conference participant, associated to a unique conference URI that identifies the session, and responsible for conference policies. The **mixer** is a network element that receives sets of media streams of a single type, combining them and redistributing them to each conference participant. The physical location of these functions can vary. They can be placed in a single centralized server, or multiple servers, and in some cases there can be a focus function carried out by a single user agent, or a mixer function distributed across all participating UAs.

2.2 Characterizing MVV sessions

Sessions with multiview video (MVV) tend to be more complex and present a wider variety of configurations than those using conventional 2D video. A 2D video call or conference (like the examples shown in section 2.1.3) can usually be described with a relatively small number of parameters: signalling protocol, number of users, and media codecs, among others. However, the addition of multiview video to a session introduces several new properties.

In this section, we identify and define a set of properties that can be used to describe a video communication system, including those that support MVV. We have divided these properties in features and configuration parameters. A **feature** is a system property that provides direct value to an end user, such as mobility, display resolution, or stereoscopic video. The set of features to be used in a session is negotiated as part of the session initiation process. A **configuration parameter** is a system property that is not directly perceived by end users, but whose value can be negotiated between user agents during session initiation.

A signalling protocol such as SIP [RFC3261] needs to be aware of these features and variables in order to properly initiate MVV sessions.

2.2.1 Property overview

A video communication system can be described with the features and configuration parameters listed in Table 1 and Table 2. Each feature or configuration parameter is further classified depending on whether it is associated with video streams (with multiview video streams as a separate category), or audio streams, or with non-media aspects of a session.

Table 1 Features of video communication systems

General features	**Multiview Video Features**
Number of participants	Stereoscopy
Number of users / terminal	Motion Parallax
Video features	Free viewpoint video
Resolution	Shared Virtual Environment
Frame Rate	**Audio Features**
Number of Displays	Audio Bandwidth
Video Scale	Spatial Audio
Video Framing	
Eye Contact	

Table 2 Configuration parameters of video communication systems

General parameters	**Multiview Video Parameters**
Signalling protocol	Multiview Video Format
Media Server	Geometric Model
Video parameters	Motion Tracking
Video Codec	Virtual View geometry
Number of Transmitted Views	**Audio Parameters**
Number of Video RTP Sessions	Audio Codec
View-Display mapping	Number of Audio Channels
	Audio Channel Geometry

2.2.2 Video communication features

2.2.2.1 General Features

We classify as general features those features of a conferencing system that are not related to audio or video processes. Our classification includes two features of this category: the number of participants in a session, and the number of users per participating terminal.

In the context of a multimedia session, a **participant** is defined as a software endpoint that connects one or more users (or an automata) to the session [RFC4353]. In SIP-based systems, every participant includes a SIP User Agent. The **number of participants** supported by a communication system determines how many SIP UAs can join a session for that system. Most video communication systems can handle 2-participant sessions (also known as video calls), whereas sessions with 3 or more participants (or video conferences) are less commonly supported, and typically require dedicated infrastructure like media servers.

The **number of users** supported by a terminal depends on factors like terminal type and screen size. Generally speaking, mobile terminals are single-user, whereas desktop devices typically have one user but can accommodate two or three users with some effort. Beyond that, the only terminals that can handle higher numbers of users in a comfortable manner are dedicated conferencing rooms. Having two or more users per terminal can make it harder to provide features like eye contact or stereoscopy.

2.2.2.2 Video Features

In this category, we include features associated to video streams. These can apply to either 2D video or multiview video.

Video resolution, or the amount of pixels that compose an image, is one of the most direct indicators of video quality. The resolution at which video is displayed in a communication system is constrained by many factors, including capture resolution, screen resolution, choice of video codec, or available bandwidth. Current conferencing systems present a wide variety of video resolutions, up to High Definition resolution (1920x1080).

Frame rate, or the number of video frames displayed per second, is another quality measure for video. The main limiting factors for frame rate in a conferencing system are processing power at the endpoints (for encoding and decoding), video codec, and available bandwidth, as well as the frame rate of capture devices. For a given set of endpoints, codecs, and bandwidth, frame rate can be improved at the cost of sacrificing video resolution or encoding quality, and vice versa. Frame rates of 24 to 30fps (frames per second) are considered good for a conferencing system, whereas optimum quality is attained at 60 fps.

The **number of displays** at an endpoint is most commonly 1, but certain devices, and conferencing rooms in particular, can feature one or more additional displays. Multiple displays are useful for rendering video from a remote site with many users, or from multiple remote sites, without resorting to downscaled images. It is also possible to have separate sets of displays for showing remote users and other types of media, such as presentations. Typical session configurations for multi-display terminals associate one video stream to each screen, so the bandwidth and processing requirements for these terminals are effectively multiplied, compared to single-display devices.

The property of **eye contact** describes the ability of a video conferencing system to convey the direction of a remote user's gaze, as well as other nonverbal cues that enhance communication. With regards to this property, there are three basic levels of quality that a system can provide: gaze error (when there is a deviation between a perceived angle of gaze and the intended gaze direction), eye contact (when users can always tell if they are being looked at), and spatial faithfulness (when users can always tell the target of remote user's attention). A detailed discussion of this property is provided in section 5.1.

The **scale** of a video is the size of a rendered image, relative to the original scene. Ideally, the scale of rendered objects in a conferencing session should be as close as possible to real size, in order to improve immersion. However, this is only possible for larger displays, so video is often downscaled to some degree due to screen size limitations. Upscaled video, on the other hand, is usually reserved for specific scenarios like a presentation to a large audience. For further discussion on video scale, see 5.1.2.

Video **framing** refers to the portion of a scene that is rendered to users. In conferencing contexts, this translates to showing remote users wholly, or in part only. The minimum practical framing for communication purposes is head framing, which shows a user's head and shoulders. On the other hand, the largest framing required for most typical video sessions is upper body framing (showing the upper body of users), since remote users are usually sitting behind a table. Only in specific cases, such as individuals making a presentation or a performance, may it be necessary to use full body framing. A more detailed discussion of video framing is provided in 5.1.2.

2.2.2.3 MVV Video Features

Certain video features only apply to systems using multiview video, like stereoscopy, motion parallax, free-viewpoint video, or virtual scenarios. We discuss these features below.

A conferencing system supports **stereoscopy** if it can render video with depth information from remote users. Regarding this feature, we distinguish three types of systems: those with no stereoscopy, those with glass-based stereoscopy, and those with glass-free stereoscopy. At the very least, a communication system needs to capture and transmit two views of a scene, or a view and a depth map, to provide stereoscopy.

The **free viewpoint video** (FVV) feature allows users to change in real time the perspective of a rendered scene. This is not particularly useful in typical conferencing scenarios where users are sitting at fixed positions and have no particular interest in changing viewing perspective. However, the feature is appealing for communication scenarios where a remote user is teaching a physical activity, as well as scenarios with virtual reality, or for streamed video content.

The requirements for FVV are highly asymmetrical, since capture and rendering can only be performed at sophisticated conferencing rooms with many capture devices and huge video processing power, but FVV can be received and displayed at any video endpoint that can send a data stream for perspective control. For this reason, we define three types of communication system regarding this property: systems without FVV, systems with symmetric FVV (i.e. FVV is sent and transmitted at both endpoints), and systems with asymmetric FVV (i.e. one side sends FVV, and the other receives FVV and sends regular video).

Motion parallax, or the change in observed viewpoint matching a viewer’s motion, is a visual cue for depth information that can complement stereoscopy. Compared to stereoscopy, it has an increased working range, and has the advantage of being usable by viewers with binocular vision deficiencies. [2007-Konrad]. Providing motion parallax requires a greater number of views than stereoscopy; these views are either generated dynamically following the input from a tracking device (see section 2.2.3.3), or are displayed simultaneously and multiplexed spatially via special, multiview displays.

In order to provide motion parallax, a system must be capable of rendering free viewpoint video, though the range of viewpoint angles required for a convincing parallax effect is much more narrow than that usually associated with true FVV. On the other hand, not every FVV system will necessarily feature motion parallax, since the rendered views need to match the user’s position. FVV Systems that use methods other than tracking for viewpoint selection, such as manual input, will lack motion parallax. Motion parallax will also be missing or deficient whenever multiple users are sharing a terminal unless different sets of views are sent to each one.

Note that stereoscopy and motion parallax are independent features that can appear separately, or be used together for an optimal depth perception. Likewise, though motion parallax is based on the same principles as free viewpoint video, it has different applications and requirements, so we will treat both as different features for the purposes of our classification.

A **Shared Virtual Environment** is a computer-generated scenario that provides a context to place each user in a video conference. Usually, this takes the form of a virtual table with participating users arranged around it. An ideal virtual environment is geometrically consistent across participating sites, so that the relative positions of objects and users remains the same for all observers – that is, it is spatially faithful (as defined in 2.2.2.2 under Eye contact, and in more detail in 5.1.1). Furthermore, in a session with a virtual environment, it is desirable that each remote user is rendered from an appropriate perspective based on the position of that user and the observer, in the virtual space of the session – which requires capturing and transmitting multiple views from each site.

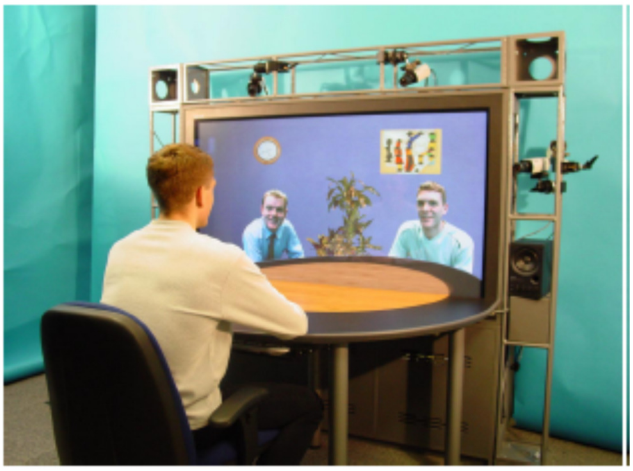

Figure 4 Example Shared Virtual Environment

[2002-Schreer]

2.2.2.4 Audio Features

In this category, we include features associated to audio streams of a communication session.

Audio bandwidth, or the range of frequencies covered by an audio stream, is a parameter with a direct impact on perceived audio quality. There are two main categories of audio bandwidth used in multimedia sessions: narrowband audio, traditionally used in telephone calls, which covers the 300 Hz – 3.4 kHz band, and wideband audio, which covers a band between 50 Hz and 7 kHz, offering a much better quality.

Spatial Audio allows the placement of sound sources at arbitrary points in space; in conferencing contexts, it is used to associate the voice of remote speakers to the position of these speakers on screen. This allows for better immersion, and to better identify who is speaking at a given time.

2.2.3 Configuration parameters

2.2.3.1 General Parameters

Under general parameters, we include those configuration parameters that are not associated to a particular media stream.

A **signalling protocol**, is the protocol used by a communication system to initiate and configure a multimedia session. The choice of signalling protocol is crucial for interoperability; in order for two endpoints to successfully communicate, they must support a common signalling protocol, or a media gateway must be available to provide translation. The most widely adopted signalling protocol for multimedia sessions is the Session Initiation Protocol, or SIP [RFC3261], standardized by the Internet Engineering Task Force [IETF]; another popular protocol is H.323 [H323], a standard from the ITU Telecommunication Standardization Sector [ITU-T].

The lack of a common signalling standard for sessions with multiview video is a major motivating factor for this document. In the following chapters, we present a series of solutions to extend SIP in order to support multiview video features.

Media servers are network nodes that intercept media streams in a multimedia session, performing certain operations on them, and forwarding them to the receiving endpoints. The most common scenarios involving media servers are conferences, or sessions with three or more participants. It is usually impractical to have each user agent in a conference send video streams to all remote participants, so instead all streams are sent to a centralized media server, which processes and redistributes them as needed. We distinguish two main types of media servers: mixing and non-mixing servers. A mixing server, or mixer [RFC4353], decodes and combines streams of a given media type, and sends it to receiving participants. Non-mixing servers, on the other hand, do not perform any media combination, but carry out other operations such as stream switching or transcoding.

2.2.3.2 Video Parameters

In this category, we include configuration parameters associated to video streams. These can apply to either 2D video or multiview video.

The **video codec** used to encode a video stream is a critical choice for a conferencing system, affecting the bandwidth and processing requirements and, consequently, determining the quality, resolution and frame rate at which video can be displayed. The codec most commonly used nowadays is the H.264/AVC standard [H264], which provides good compression rates with a reasonable level of complexity, but many alternatives exist.

The **number of views transmitted** by each endpoint in a session has an obvious impact in resource consumption, but also in the range of available features. Though most 2D video systems are designed to send and receive a single view per participant, the use of additional video streams is not exclusive to multiview systems. As an example, it is possible to have two or more video streams originating from the same terminal that are otherwise unrelated and not part of a multiview video stream, such as the streams associated with different screens in a multi-screen conference room.

A media stream transmitted over an IP network is usually encapsulated within a transport protocol, the most common of which is the Real-time Transport Protocol (RTP) [RTP]. A RTP transport stream is called a **RTP session** (not to be confused with a multimedia session; multimedia sessions usually include one or more RTP sessions), and is identified by an origin address and port, and a destination address and port. Each RTP session is processed separately by intermediate network nodes such as firewalls, media servers, or quality of service infrastructure.

The **number of video RTP sessions** usually, but not always, coincides with the number of transmitted views. It is possible to have several video views within a given RTP session – the RTP specification discourages including multiple media streams per RTP session, but there are techniques to multiplex views inside a video stream without infringing this rule. Examples of these techniques include frame-packing mechanisms, the video-plus-depth format [MPEG-C-PART3] and certain configurations of Multiview Video Coding (MVC) [MVC], all of which are associated to multiview applications.

By **view-display mapping**, we refer to the association between received video views and the displays showing them, at a given endpoint. Aside from the trivial case of sessions with a single view and a single display, there are two types of scenario where this association is relevant: terminals with multiple screens, and sessions where more than one view is shown on a given display. The first case requires that user agents negotiating a session are able to exchange information about their number and position of displays – this would allow a user agent, for example, to know that it can send one video stream for each of a remote endpoint's three displays, and which stream corresponds to the leftmost display, and so on.

As for the scenario of multiple views per screen, it can be further divided in two cases: a 2D display showing different views on specific sections of the display (e.g. one remote participant on the left half of the display, and another on the right half), or a 3D or multiview display showing two or more views, a subset of which can be observed depending on factors like use of glasses or observer's position.

These view-display mapping scenarios are summarized in Figure 5. Note that these scenarios can appear simultaneously in a single session, for example if an endpoint has more than one multiview screen.

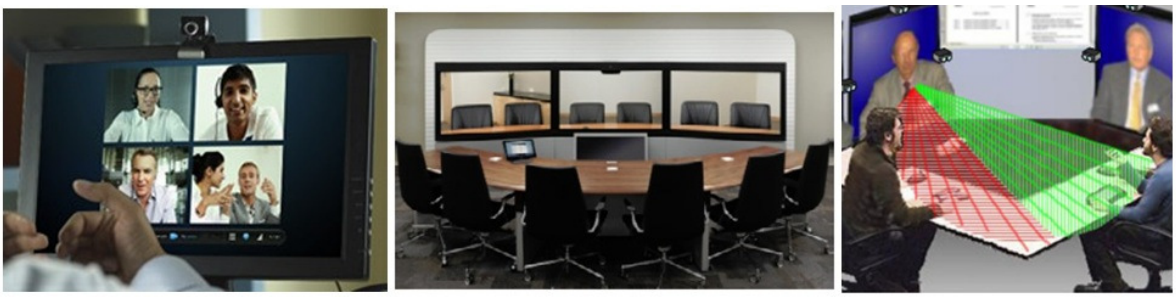

Figure 5 View-Display mapping.
Left: multiple views in 2D screen (source: www.skype.com); Center: three screen conference room (source: www.cisco.com); Right: multiview display (source: [2008-Schreer])

2.2.3.3 MVV Video Parameters

Certain video configuration parameters only apply to multiview video streams. Examples of these are the choice of multiview video format, the geometric model, the use of tracking, and the configuration of virtual views.

A multiview video stream is composed of a series of video views and depth maps, which can be distributed across one or more transport streams, and encoded together or separately. Each method for transporting, encoding, and associating the component streams of a multiview video stream is called a **multiview video format**. Examples of known formats include simulcast (encoding and transporting each view or depth map separately), video-plus-depth [MPEG-C-PART3] (encapsulating a depth map as metadata within a video stream), frame-packing [H.264] (multiplexing views and/or maps in a video stream, using spatial multiplexing or time multiplexing), and Multiview Video Coding [MVC] (encoding multiple views together using inter-view dependencies). Multiview video formats for 3D video streams are discussed in detail in chapter 3.

Many multiview video streams include some kind of **geometric model** to describe the represented 3D scene. These come in a variety of forms, such as depth maps, polygonal meshes, or voxels, among other techniques [2007-Alatan].

Motion tracking techniques allow a conferencing system to be aware of changes in the position of a user (or of specific body parts of the user) in real time. Common types of tracking include head tracking, eye tracking, hand tracking, or full body tracking. The most important application of tracking technologies in conferencing is the provision of motion parallax effects (see 2.2.2.3). Motion parallax requires changing the perspective of rendered views to match user movements, which can be implemented by using a head tracking device as input in the rendering of virtual views [2007-Konrad]. Some conferencing systems also use eye tracking to identify the object of attention of each participating user [2003-Vertegaal]. Finally, it is possible to use hand or body tracking in combination with a gestural recognition system as a user interface for a conferencing system [2011-Perez]

Note that, depending on system configuration, motion tracking data may be processed locally, or transmitted to other network nodes as a data stream.

A virtual view is a rendered video view whose perspective does not correspond to any capture device, but to an arbitrary viewpoint. We define **virtual view geometry** as the point of origin and view direction associated with a virtual view. These point and direction can be subject to real time changes over the course of a session, as is the case for virtual views in free viewpoint video applications. However, it is also possible to have multiview video applications that use static virtual views whose geometry is determined during session initiation – for providing better eye contact between participating users, for example. In a session with dynamic virtual views, each of these virtual views should have an associated control data stream to adjust its viewpoint over the course of a session.

2.2.3.4 Audio Parameters

In this category, we include configuration parameters associated to audio streams of a communication session.

The **audio codec** used to encode an audio stream determines properties like audio bandwidth, network bandwidth consumption, or encoding delay.

An audio stream may be composed of one or more **audio channels**, representing audio captured at different positions in a conferencing site.

In sessions with spatial audio, positioning information for each audio channel must be provided to receiving users; we call this information **audio channel geometry**. The position for each channel may remain static over the course of a session, in which case, it must be negotiated during session initiation. However, it is also possible to have dynamically changing geometry for audio channels, for example in a free-viewpoint video session where the perspective of a view can be adjusted, and the position of audio sources must be changed to match the observed scene. In this case, there should be a control data stream associated with the audio stream to manage these position changes.

2.3 Communication scenarios with multiview video

As we have explained in the previous section, multiview video communications can incorporate very different sets of features, and there are many configuration parameters that can vary from one system to the other. Taking this into account, it is not surprising that they can be applied on a wide variety of use cases. In this section, we identify and describe three of the most likely usage scenarios of these technologies, and characterize them using the features and parameters we have defined.

For our example communication scenarios, we have selected a stereo-enhanced video session, a tele-education session with free viewpoint video, and a immersive telepresence system using multiview video to provide better eye contact and a shared virtual environment.

2.3.1 Stereo video session

On section 2.1.3, we describe 2D video calls and 2D video conferences, two types of multimedia sessions that are well supported by current desktop and mobile communication devices. Extending such sessions to support stereoscopic video would result in **stereo video calls** or **stereo video conferences**, which add depth perception but otherwise preserve the overall user experience.

Scenario description. Two or more users want to engage in a remote video conversation using mobile or desktop terminals. These terminals have stereoscopic capture and display capabilities, so the video feeds they receive from each other are enhanced by depth information. Apart from the addition of stereo video, from the users' point of view, the process is indistinguishable from a 2D video session that could be performed over desktop or mobile terminals.

Features. For the purposes of this scenario, we will consider non-immersive calls and conferences, that can be run in domestic or mobile environments. These sessions typically have, but are not limited to, a single user per device. Feature-wise, these sessions are equivalent to their 2D counterparts, except for the addition of stereo video: average resolution and frame rate, no special requirements regarding eye contact, scale, framing, or audio properties. Table 3 summarizes the multiview features of this scenario.

Table 3 Stereo session: Multiview Video Feature overview

Multiview Video features	
Stereoscopy	Yes
Motion Parallax	No
Free-Viewpoint Video	No
Shared Virtual Environment	No

Configuration Parameters. Compared to a 2D video session, the video configuration parameters present some changes. The 3D video stream requires the transmission of 2 or more views or, alternately, of one view and a depth map. A multiview video format needs to be defined for the 3D video stream, and the views or depth maps may be transmitted as separate transport streams, or together, depending on the choice of multiview video format. **Table 4** summarizes these configuration parameters.

Table 4 Stereo session: Video configuration Parameters

Video Parameters	
Video Codec	Any
Views Transmitted	1+
Video RTP sessions	1+
View-Display Mapping	2+ Views / Screen
Multiview Video Parameters	
Multiview Video Format	Required
Geometric model	Depth Map (Optional)
Motion Tracking	No
Virtual view geometry	No

System requirements. Stereoscopic capture and display devices are needed at each terminal, but otherwise, this scenario should be able to run on a high end PC or mobile device over commercial networks. Autostereoscopic displays are advised due to their lack of glasses. The stereoscopic streams can be sent using any multiview video format (simulcast, video-plus-depth, frame-packing or MVC), and multiple streams can optionally be multiplexed over a single RTP session. In the case of a multiparty session, an intermediate media server should be used for mixing. Video mixing for the 3D video stream needs to be aware of the additional views or depth maps, so that the depth information of each party is preserved.

2.3.2 Tele-education with Free Viewpoint Video

Tele-education is one of the communication scenarios that can benefit the most from the addition of free viewpoint video (FVV). Frequently, in other types of session, free viewpoint isn't particularly useful, or even desirable, because the priority is to achieve eye contact between remote users, and this isn't compatible with arbitrary changes in perspective. However, eye contact is not as crucial for a teacher addressing a large number of students, whereas dynamic perspectives can prove valuable in the teaching of certain disciplines.

Consider a complex physical activity, such as a sport, dance, or craft. When recording such activity on a fixed-perspective video, loss of important information can occur, due to changes in the performer's orientation, or to certain steps of the process taking place on the side opposite to the camera. This makes free viewpoint video a desirable feature for this kind of teaching.

Scenario description. A teacher intends to give a lecture to students in one or more remote sites. The lesson is of practical nature, and is enhanced by free viewpoint video. The teacher's site is equipped with a multiview capture setup, and has a 2D display to show feedback from the students, who are equipped with regular 2D cameras and displays. Multiview video information is transmitted so that the student sites are capable of selecting perspectives of the captured scene within a certain operating range, and rendering them in real time.

Features. This scenario is asymmetrical, in that the media properties in each direction of communication are drastically different. The teacher's endpoint sends a free-viewpoint-video stream capturing the whole body of the teacher from a wide range of angles. Since the important part of the scene at the teacher's site is the teacher himself, rather than the background, it may be desirable to remove the background from the captured scene, and render the teacher over a virtual environment. There are no special requirements regarding video resolution, frame rate, or scale, or any audio properties. In the opposite direction, the student sites transmit plain 2D video. A summary of relevant features is shown in Table 5.

Table 5 FVV Tele-education: Video Feature Overview

Direction	**Teacher -> Student**	**Student -> Teacher**
Video features		
Video Framing	Full Body	Any
Multiview Video features		
Stereoscopy	No	No
Motion Parallax	No	No
Free-Viewpoint Video	Yes	No
Shared Virtual Environment	Optional	No

Configuration Parameters. For the media sent from student sites, configuration parameters are the same as those of a 2D video call, with one exception: these sites need to send a data stream to control the perspective of free-viewpoint video. In the opposite direction, for the media transmitted from the teacher's site, two different configurations are possible: to send the full free viewpoint video stream, or to send a single virtual view.

Sending the full FVV stream has the advantage of minimizing the delay in adjusting view perspective, but this comes at a huge cost, since it involves sending a large number of video views over the network and (optionally) a geometric model, which results in a significant bandwidth consumption. Moreover, the receiving endpoints at the student sites must be able to decode and render the free viewpoint video stream, an operation that requires considerable processing power. The bandwidth and processing requirements increase with the range of supported viewpoints, so this kind of free viewpoint video setup is only practical over local networks, or for very limited viewing ranges.

A possible alternative is for the FVV-capturing endpoint to render and transmit a single virtual view with the perspective selected by a remote user, rather than the whole FVV stream. This drastically cuts the bandwidth requirements, and allows any video terminal to receive the stream, since the virtual view is a conventional 2D video stream. On the other hand, this configuration places the burden of rendering the virtual view on the transmitting side, which may increase video delay. In addition, view selection delay is increased, since virtual views are not rendered locally on the receiving endpoint, but at the transmitting one – and the data stream for viewpoint control needs to travel through the network. Regardless, in many cases, this setup will be the only viable way of transmitting free-viewpoint video for the current scenario.

Video Configuration parameters for both scenario configurations are summarized in Table *6*

Table 6 FVV Tele-education: video configuration parameters

Direction	**Teacher -> Student**	**Teacher -> Student**
Transmission Mode	**FVV Stream**	**Single Virtual View**
Video Codec	Any	Any
Views Transmitted	Many	1
Video RTP sessions	1+	1
View-Display Mapping	None	None
Multiview Video Parameters		
Multiview Video Format	Required	No
Geometric model	Optional	No
Motion Tracking	No	No
Virtual view geometry	Required (dynamic)	Required (dynamic)

System requirements. The terminal of the teacher user consists on a dedicated conference room, with a multiview camera setup distributed so as to cover a wide angle around the teacher. Specialized hardware is required to process, encode and transmit captured multiview streams, and a network connection with high uplink capacity is needed.

On the student sites, no special display or capture configurations are required. Other requirements depend on whether the teacher site transmits a full FVV stream, or just a single virtual view (as explained above). If the single virtual view configuration is used, no special bandwidth or processing capabilities are required at the student endpoint. However, if the full FVV stream is sent, a high bandwidth for the downlink direction will be required, as well as hardware potent enough to decode and render the video.

For the single virtual view configuration, student terminals need some means to change the view selection, and request a new viewpoint from the media server. One possible way to implement this is using Dual-tone multi-frequency signaling (DTMF) [RFC4733].

2.3.3 Immersive Telepresence with Multiview Video

The term telepresence is often used to label high end videoconferencing systems that intend to provide the users with a sense of immersion. This is achieved through the use of high quality video and audio, large displays capable of presenting real-sized remote participants, and good eye contact, among other features.

Currently, commercial telepresence systems, though they can offer very good image quality and sense of immersion, are limited to the use of 2D video. However, the experience they offer can be further improved by incorporating stereoscopic video and using multiview video to enhance eye contact and gesture interactions. One such conferencing system is described in [8]

Scenario description. A group of users wants to conduct a meeting through a telepresence system, connecting to the session from several sites, and with each terminal supporting multiple users. The system is intended to provide the best possible visual quality and level of immersion, and features dedicated conferencing rooms with high definition video, and large screens that allow displaying remote users at real size. Participants are arranged around a shared virtual table, and each remote participant is shown in a separate screen, which is autostereoscopic and displays two different views for each user in the room. This allows not only to convey the sense of depth, but also to ensure that every user can see all remote participants from an adequate perspective. Because of this, this system preserves spatial faithfulness better than 2D telepresence systems, resulting in improved eye contact and gesture awareness.

Features. Table 7 summarizes the features required by this scenario. To begin with, the scenario has the same basic features as current state-of-the-art 2D telepresence systems: support for multiple participants and multiple users per participant, and excellent audio-visual quality. To that, the use of multiview video allows the addition of two major features: perfect eye contact and spatial faithfulness, and a shared virtual environment. As an option, it is possible to add motion parallax effects to further improve the experience. Finally, providing stereoscopic video is currently not possible without sacrificing visual quality, though this is more an issue with current technology than an inherent limitation.

Table 7 MVV Immersive Telepresence: Feature overview

General Features	
Number of participants	2+
Number of users	1+
Video features	
Resolution	HD

Frame Rate	30+ fps
Number of Displays	1+
Video Scale	Real Size
Video Framing	Upper Body
Eye Contact	Yes (Spatially Faithful)
Multiview Video features	
Stereoscopy	No
Motion Parallax	Optional
Free-Viewpoint Video	No
Shared Virtual Environment	Yes
Audio Features	
Audio Bandwidth	Wide
Spatial Audio	Yes

Configuration Parameters. Table *8* summarizes the configuration parameters for this scenario. Each site sends multiple video views, possibly with associated depth maps. The viewpoint of these views is adjusted during session configuration so as to match the position of observing users, and a multiview video format needs to be defined. If motion parallax is offered, head tracking mechanisms need to be in place, and data streams with the tracking information have to be transmitted.

Table 8 MVV Immersive Telepresence: Configuration Parameters

Video Parameters	
Video Codec	Any
Views Transmitted	1+
Video RTP sessions	1+
View-Display Mapping	2+ Views / Screen, multiple screens
Multiview Video Parameters	
Multiview Video Format	Required
Geometric model	Depth Map (Optional)
Motion Tracking	Head (Optional)
Virtual view geometry	Yes (Fixed)

System requirements. This scenario is very demanding from a technical point of view. At each site, several autostereoscopic displays and a multiview capture setup are required, along with hardware capable of performing all video processing, such as a cluster of high end PCs. Network connections with high speeds, both upstream and downstream, are also required at all sites.

2.4 Overview of multiview video conferencing systems

In this section, we present the current state of the art regarding video conferencing systems with multiview video technology. At the moment, we are not aware of any commercial communication product with 3D video or free-viewpoint video, or other kind of multiview features. However, this has been the subject of significant research over the last decade, with several projects resulting in experimental prototypes. We provide an overview of these prototypes in the following sections, organized in rough chronological order.

2.4.1 I3DVC

Immersive 3D Videoconferencing (I3DVC) [2002-Schreer] is a 3D videoconferencing prototype capable of establishing sessions between several single-user terminals. Developed in 2002 at the Heinrich-Hertz-Institut, it uses a shared table virtual environment, rendering correct perspective views for remote participants, and supporting motion parallax effects via head tracking.

Figure 6 shows the aspect of a virtual scene rendered by I3DVC.

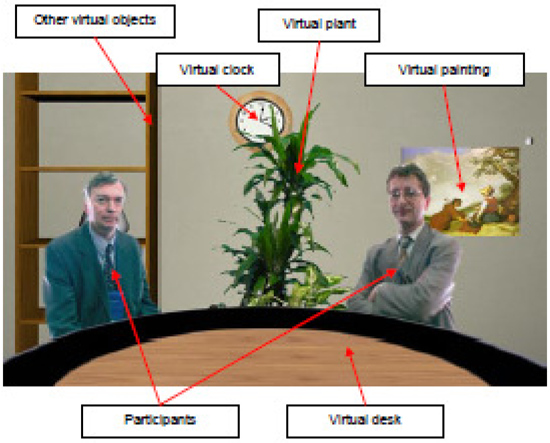

Figure 6 I3DVC Prototype

Table *9* summarizes the main features of this system.

Table 9 I3DVC Feature overview

General features	
Number of Participants	2-6
Number of users	1
Video features	
Eye Contact	Yes
Multiview Video features	
Stereoscopy	No (Optional)
Motion Parallax	Yes
Free-Viewpoint Video	No
Shared Virtual Environment	Yes

2.4.2 GAZE-2

GAZE-2 [2003-Vertegaal] is a group video conferencing system that achieves gaze awareness, by combining multiple cameras at each site with eye tracking, and control system which selects the camera that each user is looking at and broadcasts it to other participants. Participant views are presented in a virtual environment within video windows that rotate to emphasize orientation. GAZE-2 was developed in 2002 at Queen's University.

Figure *7* shows a GAZE-2 session with four users, where the left user is looking at the local user (not shown), and the center and right users are looking at the left user.

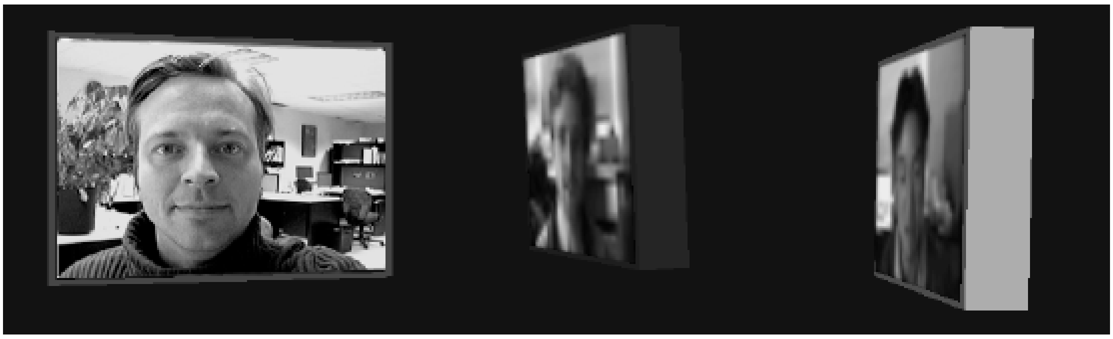

Figure 7 GAZE-2 Session

Table *10* summarizes the main features of this system.

Table 10 GAZE-2 Feature overview

General features	
Number of Participants	2+
Number of users	1
Video features	
Eye Contact	Yes (Spatially Faithful)
Multiview Video features	
Stereoscopy	No
Motion Parallax	No
Free-Viewpoint Video	No
Shared Virtual Environment	Yes

2.4.3 Blue-C

Blue-c [2003-Gross], presented in 2003 by ETH Zurich, is a high- end immersive 3D collaborative environment which surrounds the user by stereoscopic wall-sized displays, while simultaneously performing multiview video acquisition. This is achieved through projection screens that become intermittently transparent to allow for video capture, along with synchronized shutter glasses. The system is also capable of rendering 3D reconstructions of users, and have them interact and move around a virtual environment.

Figure *8* shows a user inside the blue-c terminal using a car sales application (left), and the video transmitted to the remote site.

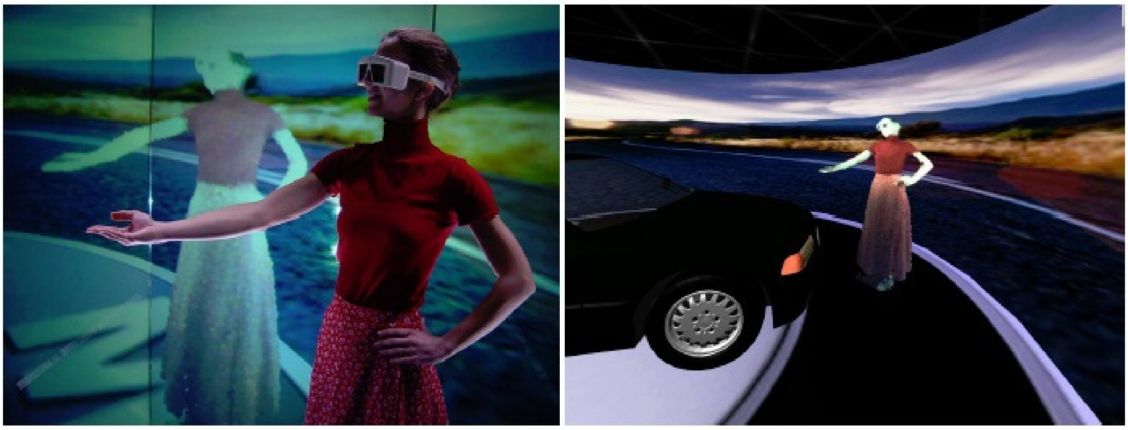

Figure 8 Blue-C application

Table 11 summarizes the features of this system. Note that eye contact is not possible within this system due to the use of shutter glasses.

Table 11 Blue-c feature overview

General features	
Number of Participants	1+
Number of users	1
Video features	
Eye Contact	No (Glasses)
Multiview Video features	
Stereoscopy	Yes (Glasses)
Free-Viewpoint Video	Yes
Shared Virtual Environment	Yes

2.4.4 Coliseum

Coliseum [2003-Baker] is a videoconferencing system designed in 2003 by Hewlett-Packard Laboratories, aimed at desktop computers connected over conventional LANs or the Internet. It is capable of presenting a virtual table, with viewpoint-appropriate 3d reconstructions rendered locally and sent through the network as 2D video. User movement in the virtual space is supported, and up to 10 participants can join a session, though frame rate degrades as the number of participants increases.

Figure *9* shows a Coliseum terminal running on a PC, and the shared virtual environment for a three-user session.

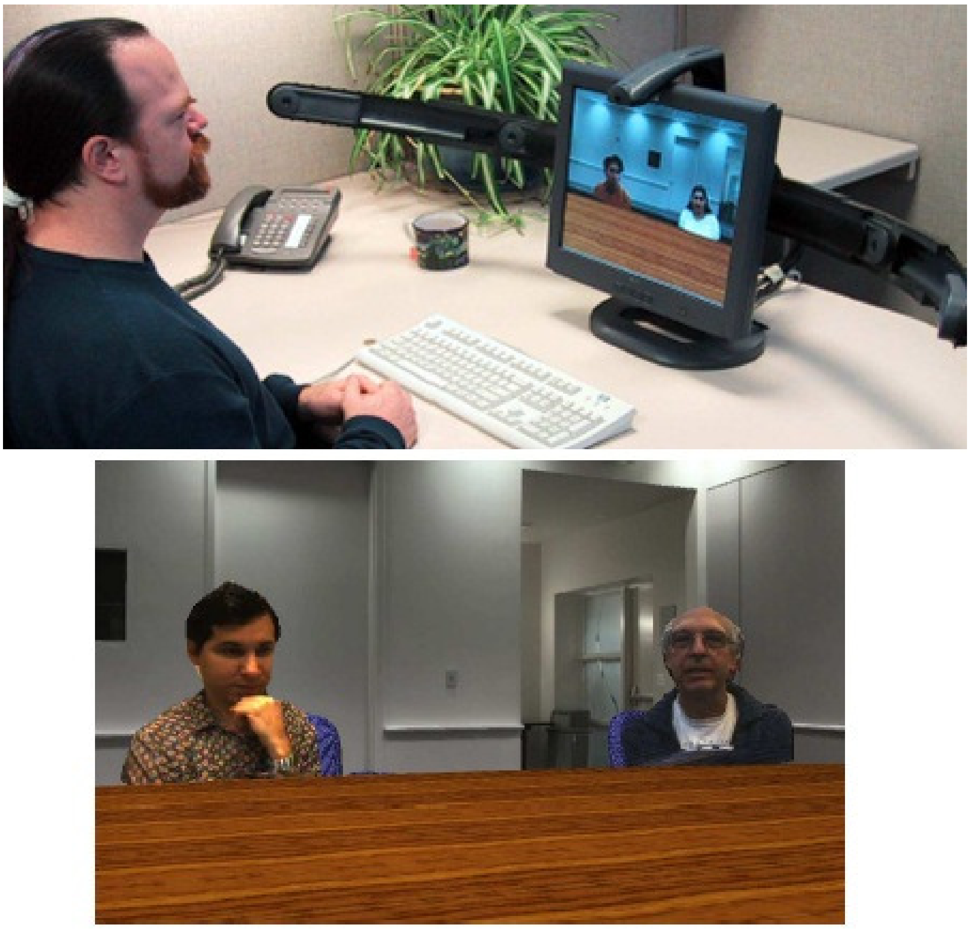

Figure 9 Coliseum terminal (top) and shared virtual environment (bottom)

Table *12* summarizes the features of Coliseum.

Table 12 Coliseum feature overview

General features	
Number of Participants	2-10
Number of users	1
Video features	
Frame Rate	Up to 15 fps
Eye Contact	Yes
Multiview Video features	
Stereoscopy	No
Motion Parallax	Yes
Free-Viewpoint Video	Yes
Shared Virtual Environment	Yes

2.4.5 MultiView

Multiview [2005-Nguyen] is a group conferencing system that addresses the challenge of keeping spatial faithfulness for all users in scenarios with multiple users per terminal. A conferencing site has a dedicated projector-based multiview display for every remote participant, configured so that each user observes a different set of views matching their position. No virtual views or virtual environment are used – rather, there are dedicated cameras capturing from each remote user's point of view. MultiView was developed in 2005 at University of California, Berkeley.

Figure *10* illustrates how MultiView works. The same display is shown from the point of view of two different users at a conferencing site; each local user can see a different perspective of the remote scene, which allows the gaze direction of remote users to be properly represented. Figure *11* shows the user, camera, and display setup for a MultiView session with three participating sites, each with three users.

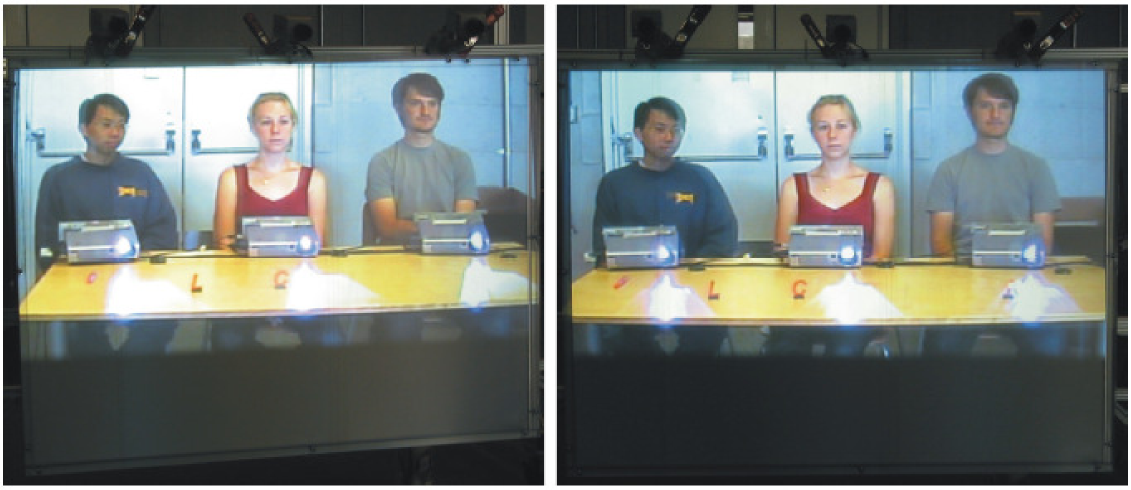

Figure 10 Snapshot of a display in a Multiview Session, from two different perspectives

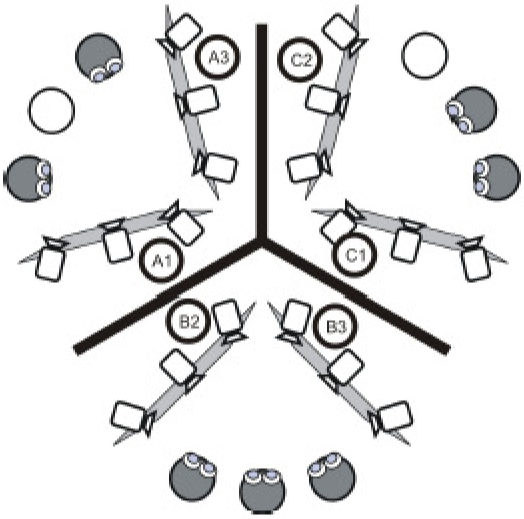

Figure 11 Setup of a three site MultiView session

Table *13* summarizes the features of this system. Note that no multiview-specific features (like stereoscopy or virtual environments) are supported by MultiView, which is fully focused on the provision of good eye contact and spatial faithfulness. On that regard, it must be noted that there is a certain vertical error in gaze direction (cameras are mounted on top of displays), of about 3º, which is acceptable according to the studies of [2002-Chen] (the reference used by the authors of MultiView), but not according to other sources (see 5.1.1.1)

Table 13 MultiView feature overview

General features	
Number of Participants	2-3
Number of users	1-3
Video features	
Eye Contact	Yes (Spatially Faithful)
Multiview Video features	
Stereoscopy	No
Motion Parallax	No
Free-Viewpoint Video	No
Shared Virtual Environment	No

2.4.6 3Dpresence

3DPresence [2009--Feldmann] is a video conferencing system that intends to combine the audiovisual quality of existing telepresence systems with multiview video features like improved eye contact, stereoscopic video, motion parallax and a shared virtual table. The system supports sessions of up to 3 participants, each with 2 users. This system is a result of European FP7 project 3DPresence, which took place between 2008 and 2010.

Figure 12 presents a conferencing site for 3DPresence, with 2 local users and 4 remote users distributed across 2 remote sites. Each remote user is shown in a separate display, and all screens have multiview capabilities to show different viewpoints depending on an observer's position. This is illustrated in Figure 13, where we can see the different images shown to the leftmost local user (left) and to the one at the right (right); note that remote users are looking at the left local user, which is correctly conveyed by the rendered views.

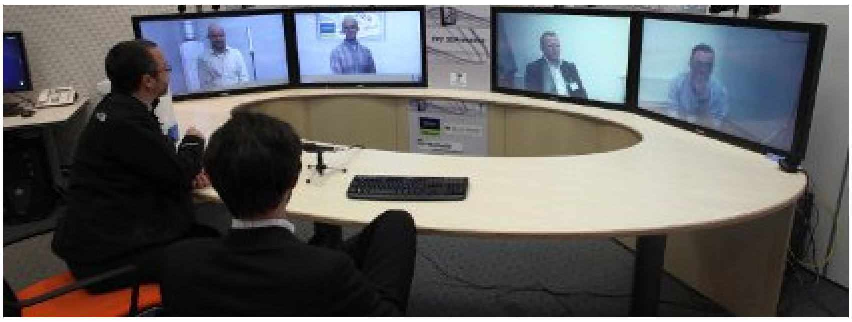

Figure 12 3DPresence conference site

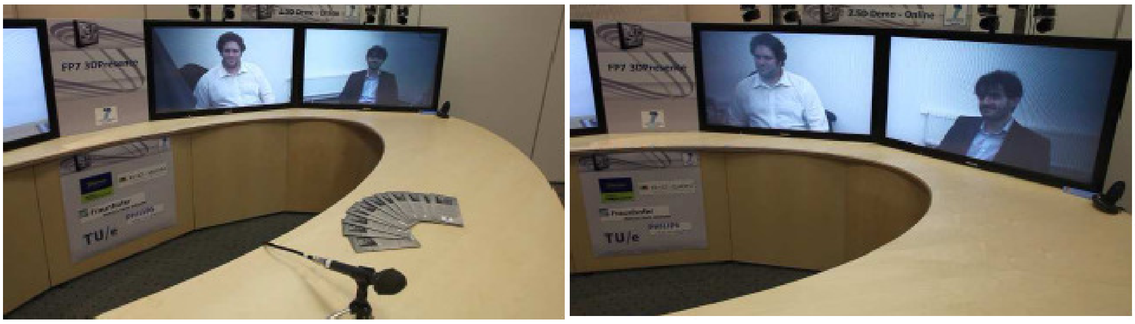

Figure 13 3DPresence multiview displays

Figure 14 Coliseum terminal (top) and shared virtual environment (bottom)

Table *14*Table *12* summarizes the features of 3DPresence.

Table 14 3DPresence feature overview

General features	
Number of Participants	3
Number of users	2
Video features	
Eye Contact	Yes
Multiview Video features	
Stereoscopy	Yes
Motion Parallax	Yes
Free-Viewpoint Video	No
Shared Virtual Environment	Yes

2.4.7 "One-to-Many 3D Video Teleconferencing System"

[2009-Jones] describes an asymmetric conferencing system where the face of one user can be presented to a remote conferencing site with multiple users over an autostereoscopic display capable of showing different views with a correct perspective to each observer. Eye contact is preserved through mirrors and a polarized camera reflected to the position of the eyes in the rendered image. In the opposite direction of communication, 2D video is used to show a view of the conferencing site with multiple users.

Figure 15 shows this system at work: at the left, the site with the 3D display, which displays a blue monochrome 3D face to multiple users. The two central sections of the figure correspond to two stereoscopic views of the remote user's face. Finally, the right of the figure shows the other conferencing site, with a 2D display showing video of the remote site and the single user is captured as a 3D video stream.

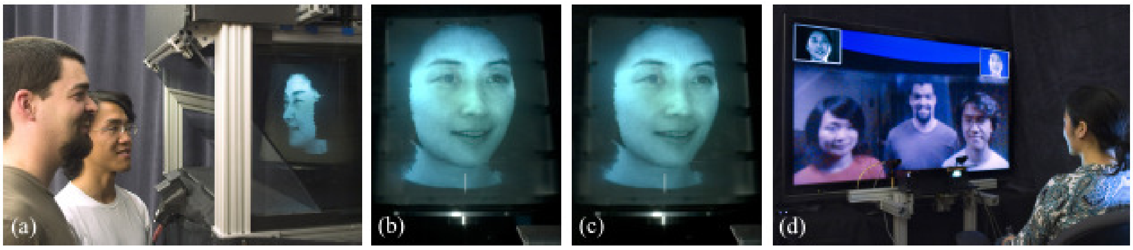

Figure 15 One-to-Many System

(a) Users looking at 3D face (b,c) stereo views of 3D face, (d), 2D video stream on remote site.

Table 14 summarizes the features of this system, for both directions of communication. Of particular note is the fact that the 3D display is monochrome – though system authors claim that color video is technically feasible, at a greater cost.

Direction	3D Capture -> Users	Users -> 3D Capture
General features		
Number of Participants	2	2
Number of users	1	1+
Video features		

Color	Monochrome	Color
Resolution	HD	HD
Frame Rate	30	30
Video Framing	Head	Upper Body
Eye Contact	Yes	Yes
Multiview Video features		
Stereoscopy	Yes	No
Motion Parallax	Yes	No
Free-Viewpoint Video	No	No
Shared Virtual Environment	No	No

Figure 16 "One-to-Many 3D Video Teleconferencing" feature overview

2.4.8 VISION

VISION [2011-Perez] is a conferencing system developed under Spanish research Project CENIT-VISION, between 2007 and 2010. The focus of VISION is to provide high definition, 3D video sessions, integrating features such as shared virtual environments, free-viewpoint video, or spatial audio. Three application profiles exist for VISION, with different use cases, sets of features and technical requirements: VISION Lite (intended for home communication scenarios), VISION Entertainment (an asymmetric free-viewpoint-video for teaching gymnastics), and VISION Corporate (a high end 3D telepresence system) .

The author of this thesis participated in the VISION project developing communication systems, which provided vital experience and insight for the current work.

Figure *17* shows a conferencing site used in VISION. Each terminal has two high definition stereoscopic displays, of which one is glass-based, and the other autostereoscopic. An array of cameras (not shown in the figure) is arranged over the room to capture a free viewpoint stream of the user from all directions. A gestural tracking system is used for interactions with the user interface.

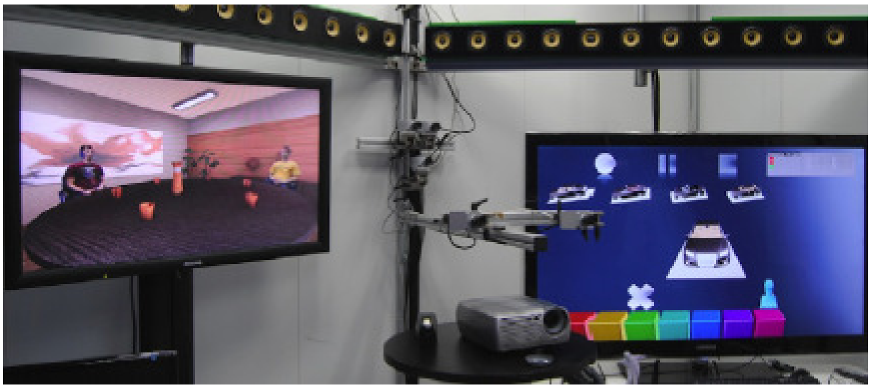

Figure 17 VISION conferencing site

Table *15* summarizes the features of the three VISION application scenarios. VISION Lite is a video call with stereoscopic video in high definition, and the option of using a virtual environment as background. In VISION Entertainment, one site sends a free viewpoint video stream to one or more remote sites, which send back a high definition 2D video stream.

Finally, VISION Corporate supports multiple participants, and has a shared virtual table which users can observe from arbitrary perspectives, as well as high definition 3D video, and spatial audio.

Table 15 VISION Feature overview

Scenario	Lite	Entertainment	Corporate
General features			
Number of Participants	2	2+	2+
Number of users	1	1	1
Video features			
Resolution	HD	HD	HD
Eye Contact	No	No	Yes
Multiview Video features			
Stereoscopy	Yes	Yes	Yes
Motion Parallax	No	No	No
Free-Viewpoint Video	No	Yes, asymmetric	Yes
Shared Virtual Environment	Yes	Yes	Yes
Audio features			
Spatial Audio	No	Yes	Yes

3 Signalling stereoscopic video in multimedia sessions with SIP/SDP

At the moment of writing this document, the existing signalling standards do not support the signalling of a multimedia session using 3D video. In this chapter, we define an extension for the Session Description Protocol (SDP) [RFC4566] to allow the use of such sessions on systems based on the Session Initiation Protocol (SIP) [RFC3261]. We begin by studying the signalling requirements that the protocol needs to meet in order to initiate a 3D video session. Based on these requirements, we provide an specification for the signalling extension.

3.1 Signalling requirements for a 3D extension of SDP

In section 2.3.1,we have specified a communication scenario called “Stereo video session”, consisting on a video call or conference in which stereoscopic video (also known as 3D video) is transmitted instead of regular 2D video, in order to display depth information to the users.

In that section, we also define a set of features and configuration parameters that characterize a 3D video session. Of these, there is one feature that is present in stereo video sessions but not in conventional video sessions (the use of stereoscopy), which needs to be indicated in signalling, but is not currently supported by SDP. In addition, there are several configuration parameters exclusive to 3D sessions, which also need to be incorporated into SDP:

- Multiview video format: whether the different views and depth maps are transmitted using simulcast, video-plus-depth, or MVC
- Number of transmitted views: if more than one view is used within a 3D video stream
- Number of RTP streams: if more than one view is transmitted per RTP stream, through multiplexing techniques such as frame packing, this must be indicated by signalling.

From these additional features and parameters, the following requirements can be defined.

3.1.1 Functional requirements

The SDP extension needs to be able to describe and negotiate the use of the stereoscopy feature in a session, as well as its associated configuration parameters. This involves the following functional requirements:

- Signal a 3D video stream composed of 2 views (left and right). The views may be transmitted as separate video streams or encoded together within a single video stream using frame-packing techniques.
- For video views and/or depth maps transmitted within a single video stream, the following options must be supported: spatial multiplexing (including side-by-side and top-bottom) and temporal multiplexing.
- Signal the presence of a depth map associated with a given video view. The depth map may be transmitted as a separate video stream from its associated view, or included within another video stream as metadata using the video-plus-depth format.
- For video views and/or depth maps transmitted as separate video streams, the association between them as a single 3D video stream must be indicated.
- Backwards compatibility: A user agent implementing the extension must be capable of initiating a video session with legacy user agents.

- Format negotiation: Two user agents supporting the extension must be able to negotiate one common 3D video scheme using the offer/answer model. Declarative usage of SDP, as used in protocols such as RTSP, must also be supported.

Note: Originally, compatibility with multiview video coding [MVC] was considered as a potential requirement. However, this functionality is already being addressed as part of an internet draft in progress[ID-MVC-RTP], and has been left out by indication of the authors of that specification.

3.1.2 Non-Functional requirements

The SDP extension also needs to meet the following non-functional requirements:

- Performance – Session initiation time, among other signalling parameters, should not be negatively affected
- Complexity – The extension should not make the signalling process excessively complex
- Extensibility – It should be possible to incorporate new 3D streaming schemes and multiplexing techniques.

3.2 Introduction

3D video applications convey depth information by showing a different view for each eye of a user. In order to achieve this, 3D video streams need to include additional information compared to conventional 2D video streams, either in the form of extra views, or as depth maps, or a combination thereof. These views and maps can be transported in a variety of ways, including, among others: as separate RTP streams (simulcast), frame-packed in a single video stream [HDMIv1.4a], or as a video stream with associated metadata (video-plus-depth).

The Session Description Protocol (SDP) [RFC4566] lacks the means to describe neither of these transport techniques for 3D video. This document extends SDP to support the description of multimedia sessions using 3D video encapsulated as simulcast streams, using frame-packing techniques, or using the video-plus-depth format [MPEG-C-PART3].

[RFC5583] defines a mechanism to signal the decoding dependency of media descriptions in SDP. This document extends that mechanism by defining a new SDP decoding dependency type, '3dd', describing the association between media streams belonging to a 3D video stream. In addition, a new SDP media-level attribute, '3dvFormat', is defined to describe the format used by media streams composing a 3D video stream. Several formats for 3D video are described in this specification, including simulcast stereo video, simulcast video and depth map, various frame-packing schemes, and streams using video-plus-depth.

3.2.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in [RFC2119].

3.3 Definitions

3D video stream: A video stream that conveys depth information by showing different perspectives of a scene to each eye of an observing user. A 3D video stream is typically composed of multiple video streams ('views'), or a combination of video streams and auxiliary data maps such as depth maps.

2D video stream: A video stream that lacks 3D depth information.

View: A video stream that represents a specific point of view of the scene in a 3D video stream.

Depth Map: An auxiliary data stream that associates a Z-value to each pixel of a view within a 3D video stream. Depth maps are often encoded as grey scale video streams.

Simulcast: A method for the transmission of 3D video streams that consists on sending additional views and auxiliary data streams as a separate RTP streams.

Video plus depth: (also known as MPEG-C Part 3): A method for the transmission of 3D video streams consisting on encapsulating a depth map as metadata within a 2D video stream. This mechanism, standardized in [MPEG-C-PART3], is compatible with MPEG-2 and H.264/AVC, and allows for backwards compatibility.

Frame-packing: A method for the transmission of 3D video streams that consists on multiplexing several views and/or auxiliary data within a single video stream, using either spatial multiplexing or time multiplexing. Frame-packing is supported by standards like [HDMIv1.4a] and [H264].

3.4 Decoding dependency of 3D video streams

The “depend” SDP attribute, defined in [RFC5583] describes the decoding dependency between two or more media descriptions. This specification defines a new dependency type for the “depend” attribute:

- 3dd: 3D video dependency – indicates that the described media stream belongs to a 3D video stream, and requires other media streams to render the 3D video. When “3dd” is used, all required media streams for the Operation Point MUST be identified by identification-tag and fmt-dependency following the “3dd” string.

Like other dependency types, 3dd is used in combination with the “DDP” grouping semantic, which is defined in [RFC5583], and based on the SDP grouping framework [RFC5888]. Whenever a 3D video stream is composed of multiple media descriptions, these media descriptions MUST be included in the same DDP group.

The media decoding dependency terminology defined in [RFC5583] can be applied to 3D video streams as follows:

- Media Bitstream: A 3D video stream is considered a Media Bitstream for the purposes of 3dd decoding dependency.
- Media Partition: Each separate media description composing a 3D video stream is considered a Media Partition. Note that each Media Partition usually contains a single video view or depth map, but can also include multiple of views/maps, e.g. when using frame-packing techniques.
- Operation Point: A subset of a 3D Media Bitstream that includes all Media Partitions required for reconstruction at a certain point of quality, number of views or depth maps, or other property. Note that a valid Operation Point for a 3D Media Bitstream can be a 2D video lacking any depth information.

3.5 The “3dvFormat”media attribute

```
a=3dvFormat:<fmt> <attribute>:<value>
```

This section defines a new media-level attribute for SDP, “3dvFormat”, which can be used to describe the transport format of a media stream in a 3D video stream. This attribute can indicate that a media description corresponds to a specific view within a 3D stream, or to a depth map, or to a combination of views or depth maps encapsulated with frame-packing techniques or with the video-plus-depth mechanism.

A media description can have multiple “3dvFormat” attributes; each attribute is mapped to a media format specified for the media, indicated by <fmt>. Only one “3dvFormat” attribute is allowed per media format.

Each “3dvFormat” attribute indicates a property (known as a “3D format attribute”) associated to a media format of its media description. The 3D format attribute consists on an attribute-value pair, with the form “<attribute>:<value>“. This specification defines four 3D format attributes: “depth-map-simulcast”, “depth-map-metadata”, “stereo-view”, “and frame-pack”.

New 3D format attributes can be defined, but they MUST be registered with IANA.

3.5.1 The “depth-map-simulcast” 3D format attribute

```
a=3dvFormat:<fmt> depth-map-simulcast:<associated_video>
```

The 3D format attribute “depth-map-simulcast” indicates that a media stream represents a depth map associated with a view within the same 3D video stream. A depth map described by this attribute is transmitted as a separate transport stream from its corresponding view.

<associated-video> is the media stream identification (the “a=mid” attribute, as defined in [RFC5888]) of the video stream associated with this depth map.

A media description with the “depth-map-simulcast” 3D format attribute MUST be included in a DDP group. This group MUST include a video stream representing the view associated with the depth map. Finally, the depth map media description MUST include a “depend” attribute with the “3dd” dependency type, indicating dependency to one or more media formats within that video stream.

Example:

```
   a=group:DDP 1 2
   m=video 1111 RTP/AVP 99
   a=rtpmap:99 H264/90000
   a=mid:1
   m=video 1112 RTP/AVP 99
   a=rtpmap:99 H264/90000
   a=3dvFormat:99 depth-map-simulcast:1
   a=mid:2
   a=depend:99 3dd 1:99
```

The example shows two media descriptions forming a 3D video stream, of which the first one (mid:1) represents a video view, and the second one (mid:2) the depth map for that view. The depth map cannot be used without its corresponding view, and this is reflected in the “depend” attribute.

3.5.2 The “depth-map-metadata” 3D format attribute

```
a=3dvFormat:<fmt> depth-map-metadata:<associated_video>
```

The 3D format attribute “depth-map-metadata” indicates that a media stream represents a depth map associated with a view within the same 3D video stream. A depth map described by this attribute is transmitted as part of the same transport stream as its corresponding view, in the form of metadata. If the view associated with this depth map is a MPEG-2 or H.264/AVC video stream, the depth map follows the format defined in MPEG-C part 3 [MPEG-C-PART3].

<associated-video> is the media stream identification (the “a=mid” attribute, as defined in [RFC5888]) of the video stream associated with this depth map.

A media description with the “depth-map-simulcast” 3D format attribute MUST be included in a DDP group. This group MUST include a video stream representing the view associated with the depth map. Finally, the depth map media description MUST include a “depend” attribute with the “3dd” dependency type, indicating dependency to that video stream.

It is important to note that, when a media format with a “depth-map-metadata” is used, the transport information for that media stream such as port, connection address or transport protocol MUST be ignored. In this case, the depth map is transmitted as part of the media stream of its associated view, rather than as a separate stream.

Example:

```
a=group:DDP 1 2
m=video 1111 RTP/AVP 99
a=rtpmap:99 H264/90000
a=mid:1
m=video 1112 RTP/AVP 99 100
a=rtpmap:99 H264/90000
a=3dvFormat:99 depth-map-simulcast:1
a=rtpmap:100 H264/90000
a=3dvFormat:100 depth-map-metadata:1
a=mid:2
a=depend:99 3dd 1:99; 100 3dd 1:99
```

The example shows two media descriptions forming a 3D video stream, of which the first one (mid:1) represents a video view, and the second one (mid:2) the depth map for that view. Two possible configurations for the depth map are offered, one using simulcast (payload type 99), and the other transmitting the depth map as metadata (payload type 100). If the depth map stream is configured as metadata, the port specified in that media description (1112) will be ignored, since the depth map will be transmitted within the video view stream. On the other hand, if the simulcast option is used, the depth map will be transmitted as a separate stream using the specified port and transport, as usual.

3.5.3 The “stereo-view” 3D format attribute

```
a=3dvFormat:<fmt> stereo-view:<view-type>
```

The 3D format attribute “stereo-view” indicates whether a video stream is associated with the left-eye view or the right-eye view of a stereo 3D video stream.

<view-type> indicates which view is associated with the media stream. It can have the value “left”, for the left-eye view, or “right”, for the right-eye view.

A media description with the “stereo-view” 3D format attribute MUST be included in a DDP group. This group MUST also include another video stream containing the “stereo-view” 3D format attribute with the other stereo view as value. The media description for either of the two stereo views MUST include a “depend” attribute with the “3dd” dependency type, indicating dependency to the stream corresponding to the other view.

Example:

```
a=group: DDP 1 2
m=video 1111 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3dvFormat:99 stereo-view:left
a=mid:1
m=video 1112 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3dvFormat:99 stereo-view:right
a=mid:2
a=depend:99 3dd 1:99
```

The example shows two media descriptions forming a stereo 3D video stream, of which the first one (mid:1) represents the left view, and the second one (mid:2) the right view. This Media Bitstream can be configured as a 3D video stream composed of two stereo views, or as a 2D video stream including just the left eye view.

3.5.4 The “frame-pack” 3D format attribute

```
a=3dvFormat:<fmt> frame-pack:<fp-format>
```

The 3d attribute indicates that frame-packing mechanisms are used in a media stream, for the specified media format.

<fp-format> signals which frame-packing mode is applied. It has three possible values: “side-by-side”, “top-bottom”, and “frame-seq”.

Of these frame-pack modes, the first two are based on spatial multiplexing, or dividing each video frame in the stream into two sub-frames, and assigning one view to each sub-frame. In “side-by-side” mode, the left sub-frame corresponds to the left eye view, and the right sub-frame to the right eye view. In “top-bottom” mode, the top sub-frame corresponds to the left eye view, and the lower sub-frame to the right eye view.

On the “frame-seq” (frame sequential) frame-packing mode, time multiplexing is used, so that half the video frames in a stream correspond to the left eye view, and the other half to the right eye view, in alternating order. In order to identify which frame corresponds to each view, additional signalling is required; in H.264/AVC video streams, this is achieved through supplemental enhancement information (SEI) metadata [H264].

3.6 Usage with SDP offer/answer model

When the extensions defined in this specification are used in the SDP offer/answer model [RFC3264], the following rules apply.

The offerer MAY include more than one “3dvFormat” attribute per media description, and the values of these “3dvFormat” can be different or duplicated. However, each media format MUST NOT have more than one “3dvFormat” attribute.

If the offerer includes a 3D video stream composed of more than one media description, all media descriptions in the stream MUST be included in a DDP group. If the 3D video stream includes streams with 3D format attributes whose description specifies any stream requirements or mandatory dependencies, those requirements or dependencies MUST be respected. Each 3D video stream in the offer SHOULD have at least one Operation Point consisting on a single 2D video stream, as well as any number of Operation Points with 3D video.

An answer MUST NOT include any “3dvFormat” attribute that is not present in the offer.

When a media format in an offered media description has a “3dvFormat” attribute, if the answer contains that media format it MUST also include the “3dvFormat” attribute, with the same parameters as the offer.

To simplify the processing of 3D video configurations, when the answer includes a “3dvFormat” attribute in a media description, the same RTP payload type number used in the offer should also be used in the answer, and the answer MUST NOT include more than one media format for that media description.

If the answerer understands the DDP semantics, it is necessary to take the “depend” attribute into consideration in the Offer/Answer procedure, as indicated in [RFC5583]

3.6.1 Backward compatibility

Depending on implementation, a node that does not understand DDP grouping or “3d” attributes SHOULD respond to an offer using this grouping or attributes either with a refusal to the request, or with an answer that ignores the grouping or 3D video format attributes.

In case of a refused request, if the offerer has identified that the refusal of the request is caused by the use of 3D video, and it still wishes to initiate a session, it SHOULD generate a new offer without any 3D video streams.

If the request is accepted but the answer is ignoring the grouping attribute, the “depend” attribute, or a “3dvFormat”, it should be assumed that the answerer is unable to send or receive 3D video streams. If the offerer still wishes to initiate a session, it SHOULD generate a new offer without any 3D video streams. Alternatively, if the answer does not include more than a single video stream, the offerer MAY initiate the session without generating a new offer, and send and receive that stream as a 2D video stream.

3.7 Examples

The following examples show SDP Offer/Answer exchanges for sessions with 3D video streams. Only the media descriptions and grouping attributes of the SDP are shown. For each example, two possible answers are considered: one in which the answering device is compatible with this specification, and one with a legacy answering device.

3.7.1 Example session with single 3D video option

The example shows a session where the 3D video stream is transmitted over a single media stream, so no grouping or decoding dependencies are needed for the SDP. The calling user agent makes a SDP offer with 2 options for configuring the 3D video stream:

- 2D video stream
- Single frame-packed video stream, with 2 views multiplexed side-by-side

Offer SDP:

```
m=video 1111 RTP/AVP 99 100
a=rtpmap:99 H264/90000
a=rtpmap:100 H264/90000
a=3dvFormat:100 frame-pack:side-by-side
```

Answer SDP:

```
m=video 2222 RTP/AVP 100
a=rtpmap:100 H264/90000
a=3dvFormat:100 frame-pack:side-by-side
```

The initial offer includes a media description with two media formats, with one corresponding to a 2D video stream(payload type 99) and the other to a frame-packed 3D video stream (payload type 100). Of these, the answering device chooses the frame-packed media format.

Alternate Answer SDP (legacy device)

```
m=video 2222 RTP/AVP 100
a=rtpmap:100 H264/90000
```

If this SDP offer is received by a legacy device and the session is not rejected, the answer will ignore any 3D video format attributes. In this case, the offerer can initiate the session treating the selected media format as a 2D video stream.

3.7.2 Test Scenario: Multiple 3D options

The example shows a session where the 3D video stream is transmitted over up to two media streams, and several options for the format of the 3D video stream are offered:

- 2D video stream
- Single frame-packed video stream, with 2 views multiplexed side-by-side
- Single video stream including a depth map as metadata
- 2 Simulcast streams, with video and depth map
- 2 Simulcast streams, with 2 stereo views.

Offer SDP:

```
a=group:DDP 1 2
m=video 1111 RTP/AVP 99 100
a=rtpmap:99 H264/90000
```

```
a=3dvFormat:99 stereo-view:left
a=rtpmap:100 H264/90000
a=3dvFormat:100 frame-pack:side-by-side
a=mid:1
m=video 1112 RTP/AVP 99 100 101
a=rtpmap:99 H264/90000
a=3dvFormat:99 depth-map-metadata:1
a=rtpmap:100 H264/90000
a=3dvFormat:100 depth-map-simulcast:1
a=rtpmap:101 H264/90000
a=3dvFormat:101 stereo-view:right
a=mid:2
a=depend:99 3dd 1:99; 100 3dd 1:99; 101 3dd 1:99
```

Answer SDP:

```
a=group:DDP 1 2
m=video 2222 RTP/AVP 99
a=rtpmap:99 H264/90000
a=3d:99 stereo-view:left
a=mid:1
m=video 2223 RTP/AVP 101
a=rtpmap:101 H264/90000
a=3d:101 stereo-view:right
a=mid:2
a=depend:101 3dd 1:99
```

The initial offer includes two media descriptions, the first of which (mid 1) can be transmitted independently, either as a 2D video stream (payload type 99) or as a frame-packed 3D stream (payload type 100). The second media description (mid 2), on the other hand, depends on the first one for all its media formats, and can be configured as a depth map transmitted as metadata (payload type 99), as a simulcast depth map stream (payload type 100), or as a right-eye stereo view (payload-type 101). The answering device chooses the configuration with 2 simulcast stereo views.

Alternate Answer SDP (legacy device)

```
m=video 2222 RTP/AVP 99
a=rtpmap:99 H264/90000
m=video 0 RTP/AVP 99
a=rtpmap:99 H264/90000
```

If this SDP offer is received by a legacy device and the session is not rejected, the answer will ignore any 3D video format attributes, as well as the grouping and dependency attributes. In the example above, the answering device has selected a media format for the first video stream, and disabled the second video stream. In this case, the offerer can initiate the session treating the selected media format as a 2D video stream. If the second video stream had not been disabled, the offerer should send a new offer with a single video stream.

3.8 Formal Grammar

The 3d attributes defined in this document use the following Augmented Backus-Naur Form (ABNF) [RFC5234] grammar.

3.8.1 “3dvFormat”media attribute

```
3dvformat-attribute = “a=3dvFormat:” fmt SP 3dvf-type
; fmt is described in RFC 4566
; fmt is media format (usually RTP payload type)

3dvf-type= depth-scast / depth-meta / st-view / f-pack
```

3.8.2 “depth-map-simulcast” 3D format attribute

```
depth-scast = “depth-map-simulcast:” identification-tag
; identification-tag is defined in [RFC5888]
```

3.8.3 “depth-map-metadata” 3D format attribute

```
depth-meta = “depth-map-metadata:” identification-tag
; identification-tag is defined in [RFC5888]
```

3.8.4 “stereo-view” 3D format attribute

```
st-view = “stereo-view:” view-type
view-type= “left” / “right”
```

3.8.5 “frame-pack” 3D format attribute

```
f-pack = “frame-pack:” fp-format
fp-format= “side-by-side” / “top-bottom” / “frame-seq”
```

3.9 Security Considerations

No security issues have been identified for this specification.

4 Signalling multiview and free viewpoint video conferencing sessions with SIP/SDP

The goal of this chapter is to propose a signalling extension for the Session Initiation Protocol (SIP) [RFC3261] to enable multimedia sessions between two or more participants featuring multiview video (MVV) or free viewpoint video (FVV). This extension will perform the following functions:

- Define a model to describe the multiview capabilities and spatial distribution of objects (users, cameras, displays) in a conferencing terminal. The signalling describes a SIP event package [RFC 3265] to allow user agents to exchange information related to this model.
- Define a model to describe the virtual space and stream map of a multimedia conference. A virtual space contains information about the spatial distribution of objects (users, cameras, displays) in the space represented by a conference, as it is shown to participating users. A stream map provides a list of all media streams in a conference, along with the origin and destination of each stream, among other parameters. The signalling describes a SIP event package [RFC 3265] to allow a conference focus to exchange information related to this model with user agents.
- Describe the complete signalling process to negotiate the configuration of a MVV conference.

Aspects of the MVV/FVV session not directly related with signalling, such as stream rendering, fall outside of the scope of this work. However, some non-signalling issues are discussed when they have direct implications on the signalling requirements.

4.1 Requirements

Before we delve into our multiview signalling solution, we need to provide some background on the types of sessions it will support, and the signalling requirements they introduce.

4.1.1 Overview of a multiview / free-viewpoint-view video conference

Two example scenarios of multimedia sessions featuring free-viewpoint-video and multiview video are presented on chapter 0: A tele-education session with free-viewpoint-video (section 2.3.2), and an immersive telepresence session using multiview (section 2.3.3). The signalling framework described in this chapter is intended to support both scenarios, along with a variety of other possible sessions based on multiview video technology. In this section, we provide a summary of common characteristics that define this kind of sessions.

For the rest of the chapter, we will use the term **MVV conference** to refer to sessions with multiview video or free viewpoint video. Note that MVV conferences usually, but not necessarily, feature 3 or more participants – though it is possible to configure a 2-participant MVV session using this signalling framework, a conference focus must be used to centralize certain signalling operations.

From a feature standpoint, the defining quality of MVV conferences is that they either offer free viewpoint video, or other MVV-dependent functionality such as spatial faithfulness enabled by virtual views. In addition, terminals in a MVV conference often (but not necessarily) support multiple local users, include more than one display, and render remote users inside a virtual space. Note that, for these conferences, stereoscopic video is considered an option, but is not otherwise given special consideration; for a detailed discussion of signalling sessions with stereo video, see chapter 0.

Regarding media configuration, the most important quality of MVV conferences is that they use multiple views per participating site, both on capture and transmission, and most often on display too. It is this abundance of media streams, not easily handled by current SIP standards, which justifies the signalling extension defined in this chapter. Additionally, other media configuration parameters are often used in these sessions and need to be taken into account by our signalling mechanism, including 3D representation method, MVV format, user tracking, and use of virtual views.

Table 16 and Table *17* summarize the main features and configuration parameters that are associated with MVV conferences but not typically present in other sessions. A detailed description of these elements can be found in Section 2.2.

Table 16 MVV Conference exclusive features

Feature	Comment
Free viewpoint video	Main feature of some MVV conferences
Shared Virtual Environment	Main feature of some MVV conferences
Eye contact	Main feature of some MVV conferences

Motion Parallax	Often present in MVV conferences
Stereoscopy	Often present in MVV conferences
Multiple users per terminal	Often present in MVV conferences
Multiple displays per terminal	Often present in MVV conferences

Table 17 MVV Conference exclusive configuration parameters

Configuration parameter	Comment
Multiple transmitted views per terminal	Present in all MVV conferences
Multiview video format	Present in all MVV conferences
Multiple displayed views per screen	Often present in MVV conferences
Geometric model	Present in most FVV conferences
Motion tracking	Often present in MVV conferences
Virtual View Geometry	Present in most FVV conferences

To illustrate the problems associated with signalling MVV conferences, let us consider a simplified version of the immersive telepresence scenario. We want to set up a multimedia conference between three SIP User Agents, set in Madrid (M) Barcelona (B), and Sevilla (S). The conferencing site associated with each UA has a single user, and two displays, each of which shows one of the remote participants. In addition, each site has two cameras that record the users from different perspectives. A special requirement of this conference is that the users should be able to see each other from specific perspectives that match what they would see if they were all present in the same room, distributed around a circular table. By showing the remote scene from each user's perspective, rather than from a single common point of view, the sense of presence and immersion is enhanced. Figure *18* shows this distribution.

To explain how this would work, let us define a **virtual space** as a common coordinate system where all conferencing sites are distributed for the purposes of this conference. This virtual space includes representations of the users, display devices, and capture devices at each site. For each of these elements, a unique identifier is provided, along with information about its position and orientation. The disposition of elements within a given site in the virtual space is consistent with how the respective users, displays and capture devices are physically distributed at the real conferencing site. Figure *19* shows the virtual space for the example conference.

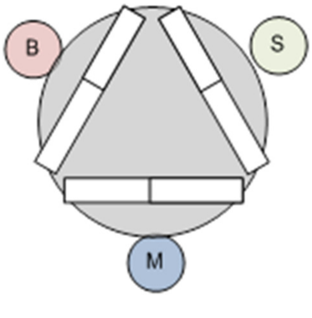

Figure 18 Example: Desired participant distribution

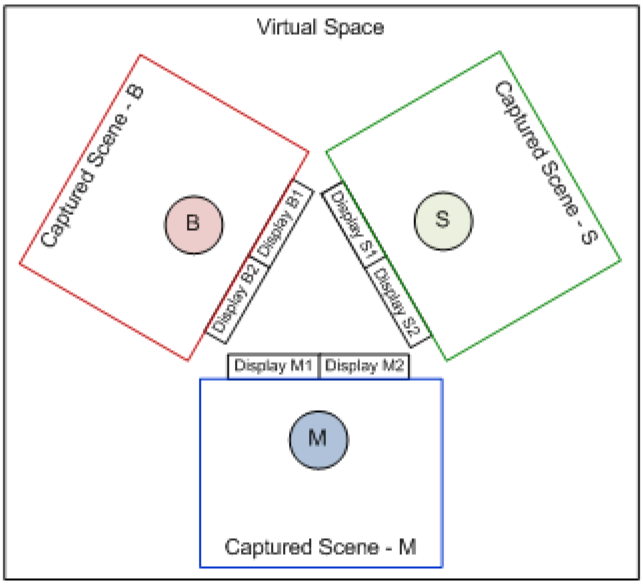

Figure 19 Participant distribution in virtual space

A virtual space defined this way can be used as a reference system to calculate the viewpoints that each user will see. In the example, each of the three users will be viewed from a different viewpoint by the two remote participants, for a total of six unique viewpoints. A viewpoint is associated with a video stream, and that stream is typically generated by a camera. In the example conference, in order to meet the perspective requirements, we want the camera for each stream to be placed at the same position in the virtual space as the user that will observe that stream. This is shown in Figure 20. In Figure 21, we can see a single site with its captured views and the corresponding cameras.

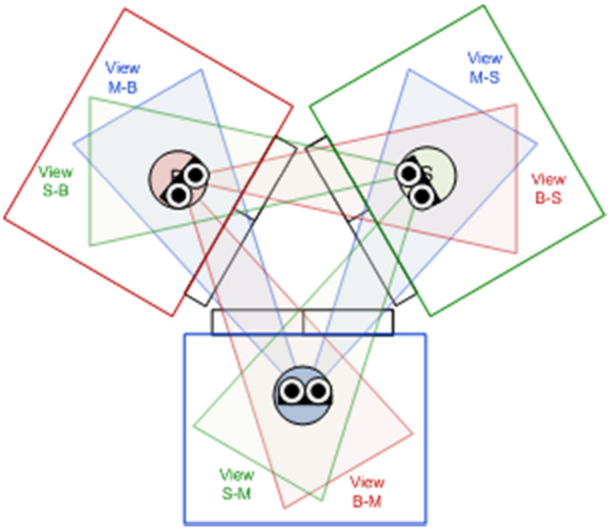

Figure 20 Viewpoints and cameras in virtual space

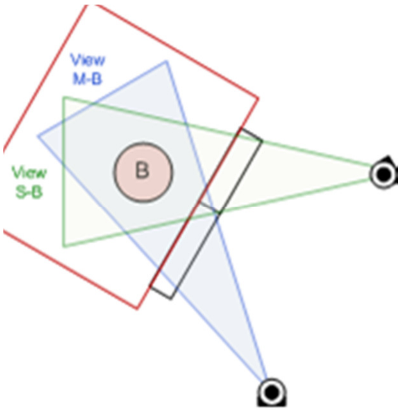

Figure 21 Physical location of cameras

Each captured video stream needs to be transmitted to a specific remote site, and rendered on the appropriate display. The unique identifier of a stream is used to map it to capture devices, receiving participants and displays. Figure *22* shows the capture and display of the video streams received by the conferencing site at Madrid, corresponding to view M-B from the site at Barcelona and view M-S from the Sevilla site.

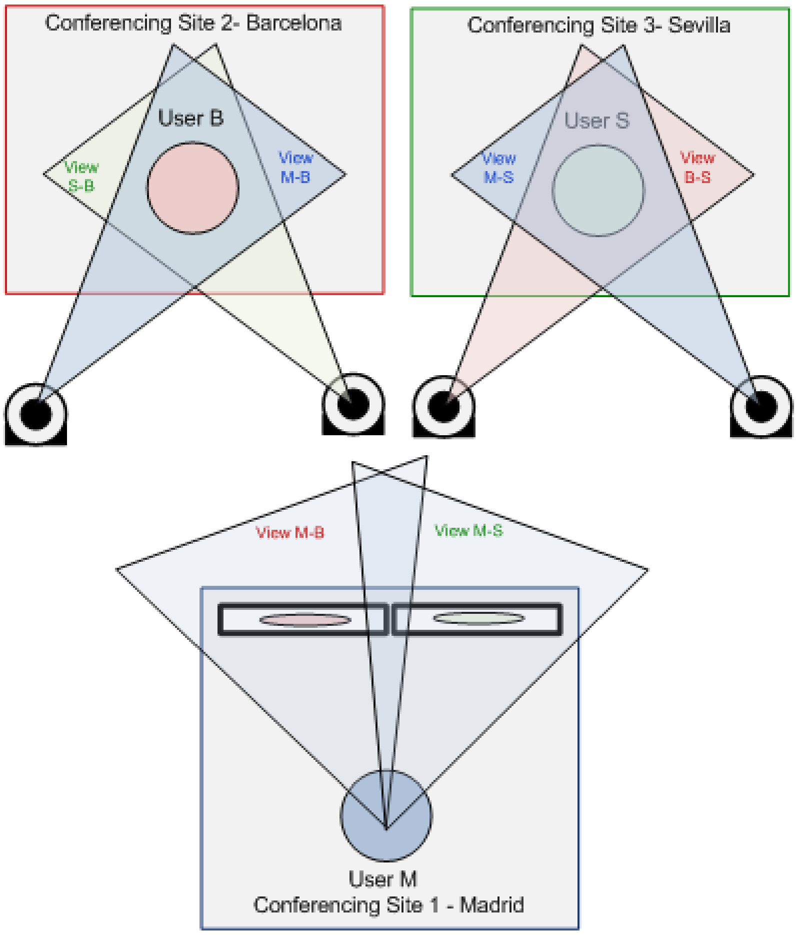
Conferencing Site 2- Barcelona
User B
View
S-B
View
M-B
Conferencing Site 3- Sevilla
User S
View
M-S
View
B-S
View M-B
View M-S
User M
Conferencing Site 1 - Madrid

Figure 22 – Capturing and displaying multiple views

To summarize, setting up this example conference wold involve the following steps:

1) Determine the spatial distribution of participants in the conference
2) Generate a virtual space matching this distribution, and including location information for users, cameras and displays at each participating site. Send a description of the virtual space to all participants.
3) Based on user and camera location in the virtual space, determine which participant will receive each captured video stream.
4) Based on user and display location in the virtual space, determine which display will render each received video stream at a site.
5) Negotiate media configuration for all media streams, including encoding parameters, receiving addresses and ports, and bandwidth configuration, among other parameters.

Of these, only 5) is properly supported by existing SIP/SDP signalling standards. The remaining points can only be addressed through a protocol extension, such as the one we propose in this chapter.

An additional challenge that should be taken into consideration for this conference is how to ensure that the camera locations match the positions of users in the virtual space. The assumption that cameras already have the proper position and orientation will only hold if all sites have been deployed specifically with this scenario in mind, and matching one another. Since this will rarely be the case, we need an additional configuration step to adjust the cameras to the required viewpoints. This can be implemented in the following ways, among others:

- Physically relocate the cameras before each conference.
- Use multiple cameras, select the one closest to intended viewpoint.
- Generate virtual views matching intended viewpoints.

The details of how to obtain a video stream matching the intended viewpoint fall outside of the scope of our work. However, from a signalling standpoint, it is important to note that these solutions share in common the need to have viewpoint information available at the sites in advance to the transmission of the video streams. In our model, this will mean that the virtual space information will have to be generated and transmitted to all sites prior to the start of the conference.

Finally, we should be aware that the last of these solutions (virtual view generation) can have a significant impact on the configuration of media streams for a conference. Virtual views are typically generated from multiple captured views. If this generation takes place at the capturing site, the resulting view behaves like a normally captured view for signalling purposes. However, if the virtual view is to be generated at the receiving end, several captured views will have to be transmitted per virtual view, resulting in a considerable increase in signalling complexity.

4.1.2 Requirements of a SIP signalling extension for MVV conferences

The main challenge that multiview video brings to multimedia conference signalling is the need for all participating parties to identify, describe and negotiate unusually large amounts of media streams. Whereas media configuration in a typical 2D video conference is mostly concerned with the parameters associated to each individual video or audio stream, for MVV conferences this is no longer the case – user agents need to be aware of the *relationships* that each stream has with other streams, with each participating site, and even with specific elements within a participating site, such as cameras, displays, or users.

Table *18* summarizes the requirements for MVV conference signalling that are not present in conventional video sessions, and thus not met by existing SIP/SDP signalling standards.

Table 18 MVV Conference signalling requirements

Requirement	
1	**Description of User Agent topology and capabilities**
1A	User agents can describe their topology.
1B	User agents can describe their multiview capabilities
1C	User Agents can exchange information about their topology parameters and multiview capabilities
2	**Description of MVV conference streams and topology**
2A	Conference topology can be described
2B	The focus of a MVV conference can generate a virtual space for the conference from the topology information of participants
2C	The mapping of media streams to user agents in a MVV conference can be described.
2D	The focus of a MVV conference can generate a map of media streams for the conference from the topology and MVV capabilities of participants
3	**Configuration of a MVV conference**
3A	User agents joining a MVV conference can use information about conference topology and stream mapping defined by the conference focus to generate SDP offers and answers
3B	The focus of a MVV conference can obtain topology and capability information from participating user agents before the start of the conference
3C	The focus of a MVV conference can generate topology and stream mapping information for the conference, and send it to participants before the start of the conference
3D	Conference topology and stream mapping can change over the course of a conference, resulting in a re-negotiation of conference configuration
3E	User Agent topology and capabilities can change over the course of a conference, resulting in an update of the conference topology and mapping of streams for that conference
3F	The conference focus can include a mixer in the conference
3G	The conference focus can include a non-mixing media server in the conference.
3H	The configuration of a MVV conference is compatible with existing network infrastructure for provision of quality of service
3I	The conference can include virtual views. The conference focus can negotiate with user agents the point of origin for these views
3J	The conference can include views with dynamically changing viewpoints
3K	The conference can include 3D video streams

The rest of the section discusses these requirements in detail.

4.1.2.1 Requirement 1: Description of User Agent topology and capabilities

There are several terminal characteristics needed for setting up a MVV conference that are not covered by existing SIP/SDP standards. The first requirement refers to the description and transmission of these characteristics. It is defined as follows.

> ***Requirement 1:*** *User Agents can describe their topology and MVV capabilities, and exchange this information with other user agents and servers.*

This requirement is divided into three sub-requirements, which are described below.

> ***Requirement 1A:*** *User agents can describe their topology.*

Topology parameters indicate the relative position of the different elements in a conferencing site, such as capture devices, displays, and users. They are used in the generation of the conference virtual space, from which viewpoints are calculated and media streams are mapped to other conference elements.

We have identified the following topology parameters:

- For each capture device:
  - Type of captured media – video, audio, or other
  - Position - usually a fixed point in space, but can also be determined during conference setup, or change dynamically during a conference
  - Area of capture – the relevant area where media is captured by this device
  - Associated users (optional) – list of users present in captured stream
- For each user in the conferencing site:
  - Position - usually a fixed point in space, but can also be determined during conference setup, or change dynamically during a conference
- For each display device:
  - Type of displayed media – video, audio
  - Position – a fixed area in space (for video displays) or point in space (for audio speakers)
  - Associated users (optional) – list of users that can observe media from this display
  - Resolution (video displays only)

> ***Requirement 1B:*** *User agents can describe their multiview capabilities.*

Multiview capabilities, such as the number of media streams that a UA can send or receive, or the maximum supported bandwidth, are used for the preparation of multipoint sessions. As we discuss in section 4.5, media negotiation in a MVV conference with multiple users may require an additional configuration step prior to SDP exchanges in SIP INVITES, to determine how many media streams will be sent and received by each participant.

We have identified the following capabilities as necessary for MVV conferences and currently unsupported by SIP:

- Maximum media streams that can be transmitted, for each media type
- Maximum media streams that can be received, for each media type
- Maximum total bandwidth transmitted
- Maximum total bandwidth received
- Media formats supported, for each media type

> ***Requirement 1C:*** *User Agents can exchange information about their topology parameters and multiview capabilities.*

At any given MVV conference, at least one network entity must be aware of all information regarding topology parameters and multiview capabilities of participating user agents. This entity will be responsible for the generation of the virtual space, and of the mapping of media streams to participants. In our signalling model, we will assign this role to the conference focus (as defined in [RFC 4353]).

Figure 23 shows the multiview capability and topology information of a SIP UA, corresponding to the Barcelona conferencing site from the example conference in 4.1.1. Note that input and output audio devices have been omitted from the figure.

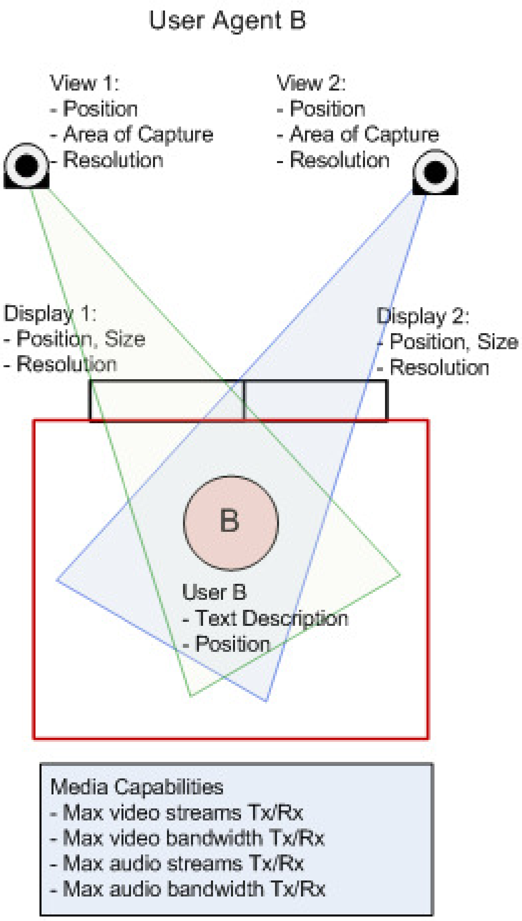

Figure 23 - Topology parameters and multiview capabilities of a SIP UA

4.1.2.2 Requirement 2: Description of MVV conference streams and topology

In order to initiate a MVV conference, participating nodes need to exchange configuration information beyond the media parameters negotiated under the SIP Offer/Answer Model [RFC 3264]. This requirement states the need to define that information and transmit it to conference participants.

> ***Requirement 2:*** *The focus of a MVV conference can define the conference topology and mapping of streams to user agents, and transmit this information to other nodes in the conference.*

This requirement is divided into sub-requirements, which are described below.

> ***Requirement 2A:*** *Conference topology can be described*

For each participating site at a MVV conference, it is possible to describe all local and remote elements in the conference (such as capture devices, displays, and users) by defining a **conference virtual space** for that participant. A conference virtual space of a conferencing site includes positioning information of these conference elements, as observed by users in that site.

By default, each participating site has a different conference virtual space, though it is desirable to configure conferences so as to make all virtual spaces as similar as possible. However, it is possible to have a single common virtual space for the conference, if the conference is spatially faithful (see section 5.1.1). Also, there are systems capable of showing different scenes to each user at a conferencing site (such as the sites with multiview displays described in 5.4); in these scenarios, there will be a different conference virtual space for each user at the site.

We have identified the following topology parameters:

- For each capture device:
  - Type of captured media – video, audio, or other
  - Position - usually a fixed point in space, but can also be determined during conference setup, or change dynamically during a conference
  - Associated users (optional) – list of users present in captured stream
  - Area of capture – the relevant area where media is captured by this device
  - Associated User Agent
- For each user:
  - Position - usually a fixed point in space, but can also be determined during conference setup, or change dynamically during conference
  - Associated User Agent
- For each display device:
  - Position – a fixed area in space (for video displays) or point in space (for audio speakers)
  - Resolution (video displays only)

    - Associated users (optional) – list of users that can observe media from this display
    - Associated User Agent

__Requirement 2B:__ The mapping of media streams to user agents in a MVV conference can be described.

In a MVV conference, each participating UA transmits a subset of its captured streams to remote participants. In order to set up the conference, all UAs need to know which streams they have to send, and to which destinations. Conversely, they also have to be aware of which streams they will receive, and their origin.

Unlike conference virtual spaces, there is always a single media stream map defined for an entire conference, which is common for all participating network elements.

To summarize, the following parameters are needed:

- For each UA
    - Transmitted streams, with
        - Local identifier of each stream
        - Destination UA (s)
    - Received streams, with
        - Media type of each stream
            - Supported media formats (optional) list of media formats that can be used for the stream
        - Associated users (optional) – list of users present in the stream
        - Associated displays (optional) – list of displays rendering the stream
            - Display coordinates (optional) – area of display where the stream will be rendered
        - Origin UA

__Requirement 2C:__ The focus of a MVV conference can generate a virtual space for the conference from the topology information of participants.

__Requirement 2D:__ The focus of a MVV conference can generate a map of media streams for the conference from the topology and MVV capabilities of participants.

A network node with access to the topology information of all user agents participating in a MVV conference should be able to generate a virtual space for the conference, as well as a media stream map with the streams that will be exchanged in the conference.

The conference virtual space should meet the following properties:

- All elements from local virtual spaces exist in the common virtual space
- Users can see each other: the arrangement of users, capture devices and display devices in the common virtual space allows each user to see all remote users.

The media stream map should meet the following properties:

- The MVV capabilities of each UA are respected: the streams transmitted or received by a UA cannot exceed the maximum bandwidth or maximum number of streams.

In addition, the following properties are desirable, and depend on both the conference virtual space and the media stream map. They should be met, if possible:

- Users in the conference have eye contact with each other (see section 5.1.1)
- The conference is spatially faithful (see section 5.1.1)
- Video of remote sites is life-sized (see section 5.1.2)

4.1.2.3 Requirement 3: Configuration of a MVV conference

The previous requirements state the need for user agents to describe and exchange information about their own topology, and that of the conference space. These steps need to be integrated into the conference negotiation process in order to configure a MVV conference:

> ***Requirement 3:*** *The configuration of a MVV conference can be negotiated between the conference focus and participating user agents, taking into account the local topology of each user agent.*

This can be further decomposed into 11 sub-requirements, which are described below.

> ***Requirement 3A:*** *User agents joining a MVV conference can use information about conference topology and stream mapping defined by the conference focus to generate SDP offers and answers.*

The main purpose of the description of the conference virtual space and the mapping of streams to receiving user agents, as defined in 4.1.2.2, is their use as input for SDP generation. This is due to MVV conference complexity: whereas in a typical SIP session only a couple media streams are usually exchanged per participant and the generation of SDP offers and answers is straightforward, that is no longer the case for MVV conferences. Rather, there can be one or more remote users, and the set of streams to send and receive can vary for each remote user and change from one conference to the other. Thus, user agents cannot independently determine which streams to include in their session descriptions, so they need a central entity (the conference focus) to provide them this information (in the form of conference topology and stream mapping).

> ***Requirement 3B:*** *The focus of a MVV conference can obtain topology and capability information from participating user agents before the start of the conference.*

> ***Requirement 3C:*** *The focus of a MVV conference can generate topology and stream mapping information for the conference, and send it to participants before the start of the conference.*

The fact that SDP generation depends on the availability of conference topology and stream mapping information introduces a constraint for the distribution of this information: it must be generated and exchanged before any participant joins the conference using SIP INVITE. Furthermore, since conference topology and mapping depend, in turn, on the availability of information for participant topology and capabilities at the conference focus, that information also needs to be exchanged prior to any SIP INVITEs.

> ***Requirement 3D:*** *Conference topology and stream mapping can change over the course of a conference, resulting in a re-negotiation of conference configuration.*

Conference topology and stream mapping are not supposed to be static documents, but need to adapt to changes such as the addition of removal of conference participants. Whenever these changes take place, all agents in the conference need to be notified of the modifications to topology and mapping. In these scenarios, conference configuration must be updated to account for these changes, and a new offer/answer exchange is needed for each remaining participant.

> ***Requirement 3E:*** *User Agent topology and capabilities can change over the course of a conference, resulting in an update of the conference topology and mapping of streams for that conference.*

Related to the previous point, the description of an user agent's topology and capabilities is also a dynamic document that is affected by events such as a user entering or leaving a conferencing site. User Agents should propagate these changes to the conference focus during a conference, which can result in updates to the conference topology and stream mapping and, eventually, a re-negotiation of conference configuration.

> ***Requirement 3F:*** *The conference focus can include a mixer in the conference.*

> ***Requirement 3G:*** *The conference focus can include a non-mixing media server in the conference.*

SIP sessions with more than two participants often incorporate mixer elements to handle media stream. [RFC4353] defines a mixer as an element that "receives a set of media streams of the same type, and combines their media in a type-specific manner, redistributing the result to each participant" . This also applies to MVV conferences, so the signalling process for these conferences must be compatible with mixers.

That said, it may not be desirable to use media mixing for certain streams in a MVV conference. Video streams in particular are often better displayed without modification. In these scenarios, a video mixer is not required, but having some intermediate element in the media path to replicate and route streams to their assigned receiving participants can still be useful. This element, which we call a non-mixing media server, must also be supported by MVV conference signalling. Non-mixing media servers do not process the content of media streams, operating only at RTP level; an example application of this kind of server in a MVV conference can be found in [2011-Perez].

Note that it is possible to combine both mixers and non-mixer servers in the same conference, for example by assigning a mixer for audio streams and sending video through a non-mixing server.

> ***Requirement 3H:*** *The configuration of a MVV conference is compatible with existing network infrastructure for provision of quality of service.*

Since they involve the exchange of many media streams at each site, MVV conferences can have consume significant network resources, and their performance degrades considerably in the presence of packet losses or high delays. Thus, it is highly desirable to be able to use quality of service (QoS) techniques with these conferences, when QoS is supported by the network. For this reason, MVV conference signalling should be compatible with current QoS infrastructure for SIP-based networks.

Requirement 3I: *The conference can include virtual views. The conference focus can negotiate with user agents the point of origin for these views.*

Virtual views are video streams that do not correspond to any single capture device, but are generated from multiple captures and are positioned at arbitrary viewpoints. These views are used in MVV conferences as a way to improve eye contact, among other applications. The conference signalling must be able to deal with virtual views, and allow a conference focus to negotiate their point of origin.

Requirement 3J: *The conference can include views with dynamically changing viewpoints.*

In some conferences, and particularly those corresponding to free-viewpoint video applications (see 2.1.1, 2.3.2), there are video streams whose viewpoint does not remain static, but can change over time (moved by user input, input from tracking devices, or other means). The conference focus and participating user agents must be able to negotiate the use of these streams.

Requirement 3K: *The conference can include 3D video streams.*

Stereoscopic 3D video can be transmitted in a variety of ways, discussed in depth in chapter 0. A MVV conference must be able to handle stereoscopic video streams offered by participants.

4.2 Solution summary

The signalling solution we present for conferences with multiview video involves two additional configuration exchanges on top of the usual SIP INVITE dialog needed to initiate a multimedia session. First, the user agents that will participate in the conference send their multiview and topology information to a user agent or application server acting as conference focus. This is performed through a SIP SUBSCRIBE/NOTIFY exchange using a newly defined event package for SIP called multiview-information, described in section 4.3.

Using this information, the conference focus generates a Multiview Conference Information document that maps which media streams will be transmitted and received by each user agent, and provides a position for each user and media source over a common virtual space. This document is then sent to the participating user agents with a second SUBSCRIBE/NOTIFY transaction. Finally, a SIP INVITE dialog is performed to negotiate the media configuration for the streams of each user agent.

Figure 24 illustrates this signalling process for an example MVV conference between two SIP user agents A and B, and an application server acting as conference focus and MCU. A detailed explanation of the steps of this process and of each signalling message is provided in the following sections.

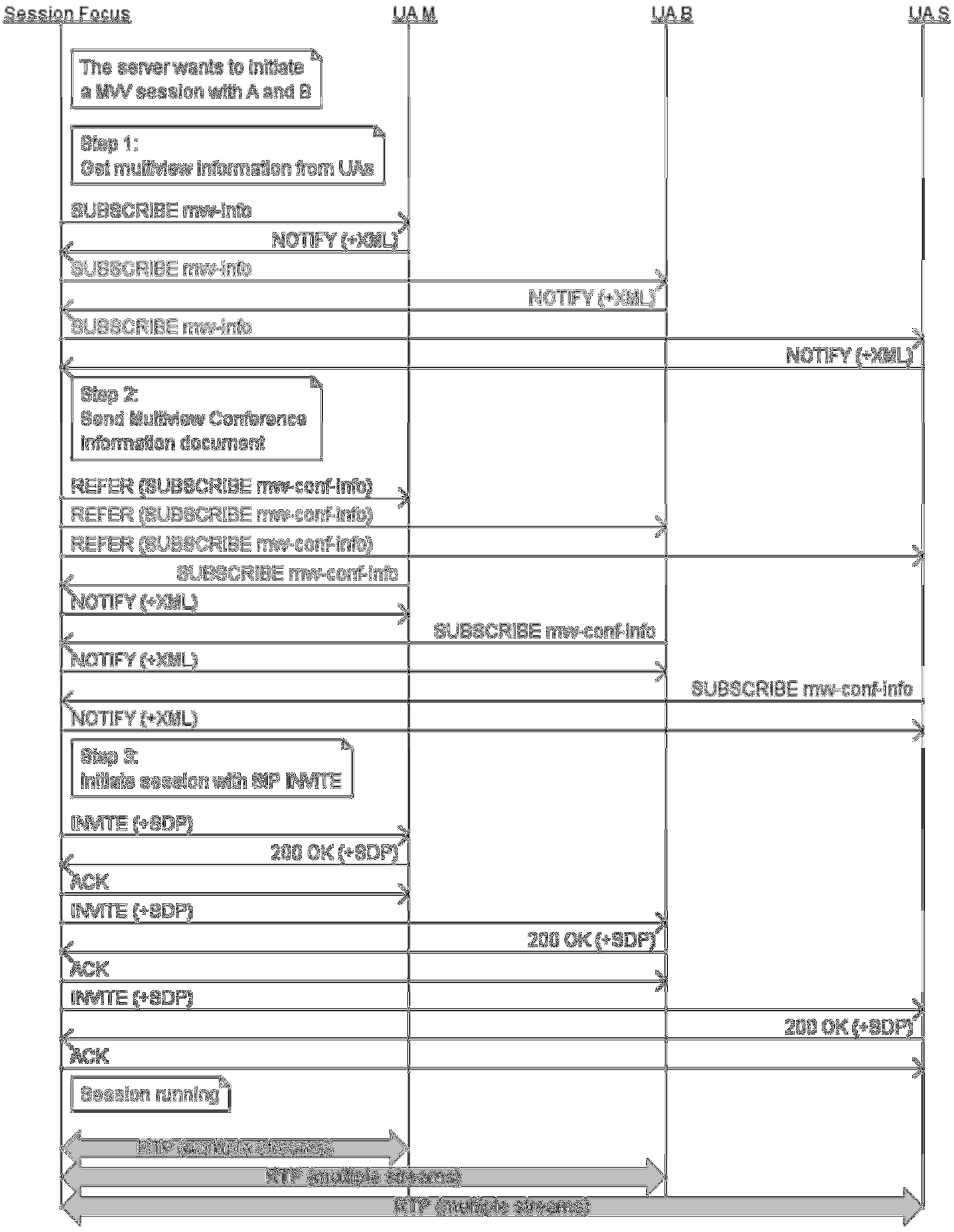

Figure 24 – Summary of signalling solution for MVV conference

4.3 Describing MVV capabilities of a UA.

In a typical SIP-based multimedia session, participating user agents only need to negotiate media parameters that can be conveyed using SDP, that is, a series of media streams and their respective configuration parameters.

However, multiview video conferences depend on information that falls outside the scope of a session descriptor, like terminal topology (i.e. spatial information of objects and users associated to the terminal), or terminal capabilities (like supported bandwidth or number of streams). In this section, we propose a XML document format for describing this information and capabilities, and a mechanism for user agents to exchange this document using SIP.

4.3.1 Multiview Information Document.

The multiview video information document (mvv-info) is an XML document associated with a SIP user agent, describing the aspects of this UA that are relevant to MVV conferences, including topology parameters and multiview capabilities. A list of these parameters and capabilities can be found on section 4.1.1 , and a more detailed discussion of each one is provided below.

The following diagram gives an example of the structure used by the mvv-info. The document includes a list of multiview capability parameters, a list of captured streams, a list of users in the conferencing site, and a list of available displays. Document elements that can appear in multiples are marked with a '*', and optional elements are enclosed in square brackets: '[ ]'.

mvv-info

- [mvv-capabilities]
    - [max-tx-bw]
    - [max-rx-bw]
    - [max-tx-streams] *
    - [max-rx-streams] *
    - [supported-formats] *
        - encoding
- capture-list
    - capture *
        - media -type
        - position
            - point
            - [position-range]
            - [position-stream-id]
            - [control-stream-id]
        - capture-area
            - [capture-range]
        - [max-bw]
        - [src-id]
        - [associated-users]
- [user-list]
    - user *
        - position
        - [description]
- [display-list]
    - Display *
        - media-type
        - position
        - [associated-users]

The rest of this section describes the elements of the document.

4.3.1.1 <mvv-info>

The multiview video information document has the root element <mvv-info>. The following mandatory attributes have been defined for <mvv-info>: "entity", and "version". In addition, mvv-info elements have an associated XML namespace name: http://jungla.dit.upm.es/~capelastegui/mvv-info . This namespace is declared using an "xmlns" attribute, as described in [XML-NS], and can be used as a default namespace or associated with a namespace prefix.

- **entity**: This attribute contains the SIP URI of the user agent described in the document.
- **version**: This attribute is an integer that increases by one between subsequent document updates. This allows the use of "state delta" processing of multiview information documents, as described in [RFC3265].

4.3.1.2 <mvv-capabilities>

<**mvv-capabilities**> is an optional element that includes information about the number and bandwidth of media streams that the described user agent can send and receive. It can include the following child elements:

- <**max-tx-bw**>: Maximum total session bandwidth, in kilobits per second, of all media streams transmitted by the user agent. The session bandwidth of a media stream corresponds to the value of the 'b=' field in SDP [RFC 4566] , when the bandwidth type for that field is 'AS' (application specific).
- <**max-rx-bw**>: Maximum total session bandwidth, in kilobits per second, of all media streams received by the user agent. The session bandwidth of a media stream corresponds to the value of the 'b=' field in SDP [RFC 4566] , when the bandwidth type for that field is 'AS' (application specific).
- <**max-tx-streams**>: The maximum number of media streams transmitted by the user agent. Optionally, this element can have a 'media-type' attribute whose value is one media type registered for "media" for SDP [RFC 4566], such as 'audio', 'video', or 'application'. If the 'media-type' attribute is present, this element indicates the limit of media streams of the specified type on the user agent; otherwise, it indicates the limit of streams of any type. Multiple <max-tx-streams> elements can be present within a <**multiview-capabilities**> element, as long as no two of them have the same 'media-type' value. If a maximum number of streams is not specified for a given media type, a default value of 1 is assumed. If a maximum number of streams for all media types is not specified, the default value is the lowest of 2 or the sum of maximum streams specified for each media type.
- <**max-rx-streams**>: The maximum number of media streams received by the user agent. Optionally, this element can have a 'media-type' attribute whose value is one media type registered for "media" for SDP [RFC 4566]. If the 'media-type' attribute is present, this element indicates the limit of media streams of the specified type on the user agent; otherwise, it indicates the limit of streams of any type. Multiple <max-tx-streams> elements can be present within a <multiview-capabilities> element, as long as no two of them have the same 'media-type' value. If a maximum number of streams is not specified for a given media type, a default value of 1 is assumed. If a maximum number of streams for all media types is not specified, the default value is the lowest of 2 or the sum of maximum streams specified for each media type.
- <**supported-formats**>: A list of encoding names (e.g. RTP payload types) supported by the terminal. It includes one or more <encoding> elements with a 'media-type' attribute containing the media type of the encoding (usually 'audio' or 'video) and a 'name' attribute containing the encoding name. It can also have an optional attribute 'fmtp' including a string with format-specific parameters used for the encoding (as would be included in a 'a=fmtp:' attribute in SDP).

4.3.1.3 <capture-list>

The element <capture-list> represents the list of media streams captured and offered by the user agent. It includes one or more <capture> elements with an 'id' attribute containing a unique stream identifier. Each <capture> can have the following child elements:

- <**media-type**>: The media type of the media stream. Must be one media type registered for "media" in SDP [RFC 4566], like "audio" or "video".

- <**position**>: The position of the capture device assigned to this stream. It has an optional 'position-type' attribute, which can have three possible values: "fixed", "variable", and "dynamic". "fixed" is the default value, used when the capture position is a fixed point in space. Captures with "variable" position can be located in an arbitrary position within a certain range, which is determined during conference configuration. Captures with "dynamic" position can be located in an arbitrary position within a certain range, which can change dynamically over the course of a conference. A <position> can have the following child elements:
  - <**point**>: An element with three attributes 'x', 'y', and 'z' representing coordinates, in millimetres, within the Cartesian coordinate system defined for the user agent. If the <position> has a dynamic position type, this element can be omitted, otherwise, it is mandatory.
  - <**position-range>:** An element representing the range of possible capture positions, when the position type is variable or dynamic. It contains several <point> elements defining the boundaries of the position range. A <position-range> can have 2 <point> elements (defining a line), 4 <point> elements (defining a quadrilateral), or 8 <point> elements (defining an hexahedron). This element must be omitted if the <position> has a fixed position type, otherwise it is mandatory. It is possible to have multiple <position-ranges> within a <position> element. In this case, all points included in these <position-ranges> are valid capture positions.
  - <**position-stream-id>:** An element representing a unique stream identifier for an auxiliary data stream associated to this capture, used whenever the <position-type> is dynamic. This stream must be transmitted as part of the same conference as the capture, and transmits the capture position in real time. The format of auxiliary streams for position information is not defined, and falls outside of the scope of this document.
  - <**control-stream-id>:** An element representing a unique stream identifier for an auxiliary data stream associated to this capture, used whenever the <position-type> is dynamic. The purpose of this auxiliary stream is to control in real time the capture position of the dynamic video stream. Unlike other media streams in this document, this stream is received by the user agent; it must be transmitted by a remote user agent receiving the dynamic video stream. The format of auxiliary streams for position control is not defined, and falls outside of the scope of this document.
- <**capture-area**> : The area of space captured in the stream. It includes four <point> elements, each with three attributes 'x', 'y', and 'z' representing coordinates in millimetres. The four <point> elements should be coplanar and form a quadrilateral. Figure 25 illustrates the area of capture of a camera. Alternately, when the <position> element of this capture has a "position-type" value of "variable" or "dynamic", these point elements can be replaced by a <capture-range> element
  - <**capture-range>:** An element representing the range of possible positions for a capture area, when the stream has a dynamic position type. A <capture-range> can have 8 <point> elements (defining an hexahedron). The actual capture area is defined in an auxiliary data stream - see the description of the

<position-stream-id> child element for the <position> of the capture. It is possible to have multiple <capture-ranges> within a <capture> element. In this case, all points included in these <capture-ranges> are valid capture area positions – however, all points in a capture area need to fit within a single <capture-range>

- **<src-id>:** The identifier for the media source for this stream. For a RTP [RFC 3550] stream, this corresponds to the SSRC value.
- <**max-bw**>: Maximum session bandwidth, in kilobits per second, of this media stream. This corresponds to the value of the 'b=' field in SDP [RFC 4566] , when the bandwidth type for that field is 'AS' (application specific).
- <**associated-users**>: An optional element listing the users captured in this media stream. If absent, all users at the site are assumed to be included in the stream. It includes one or more <user> elements with an 'id' attribute containing a unique user identifier, which must match the 'id' of a <user> element in the <user-list> of this document.

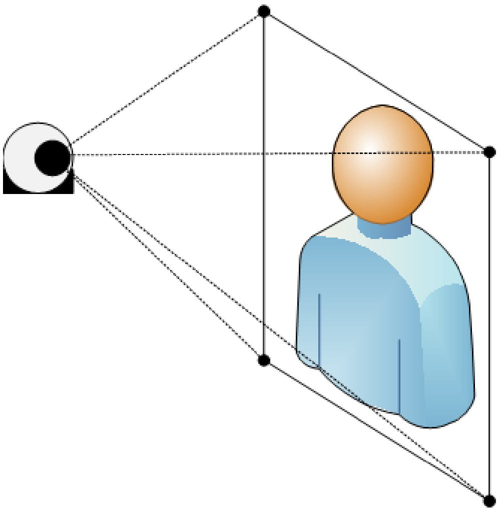

Figure 25 Area of Capture

The <src-id> and <max-bw> elements are optional, and the rest are mandatory.

4.3.1.4 <user-list>

The optional element <user-list> represents the list of users in the terminal or conferencing site associated with the user agent. It includes one or more <user> elements with an 'id' attribute containing a unique user identifier. Each <user> can have the following child elements:

- <**position**>: The position of the user. It includes one <point> element with three attributes 'x', 'y', and 'z' representing coordinates, in millimetres, within the Cartesian coordinate system defined for the user agent. This position corresponds to the middle point between the user's eyes (see Figure 26), which provides a good representation of the user's viewpoint.

- <**description**>: Optional element with a textual description of the user. This will typically include the user's name or role.

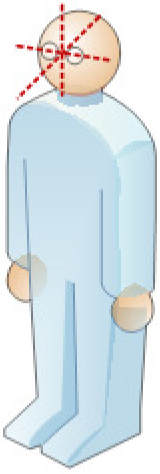

Figure 26 - User position

Note that only fixed user positions are considered. It is possible to have conferences where user positions change dynamically, but that scenario is not supported by this specification, and left for a future extension.

4.3.1.5 <display-list>

The optional element <display-list> represents the list of displays for this user agent. Display information is required for conferences that feature real size video, among other applications. A <display-list> includes one or more <display> elements with an 'id' attribute containing a unique display identifier. Each <display> can have the following child elements:

- <**media-type**>: The media type displayed by this device. Must be one media type registered for "media" in SDP [RFC 4566], like "audio" or "video".
- <**position**>: The position of the display. It includes either one or four <point> element, each with three attributes 'x', 'y', and 'z' representing coordinates, in millimetres, within the Cartesian coordinate system defined for the user agent. The four <point> elements correspond to the corners of the viewable area of the display, and should be coplanar and form a quadrilateral. Figure 27 illustrates the points defining the position of a display.
- <**associated-users**>: An optional element listing the users that can observe media from this display. If absent, all users at the site are assumed to be able to observe the displayed media. It includes one or more <user> elements with an 'id' attribute containing a unique user identifier, which must match the 'id' of a <user> element in the <user-list> of this document.

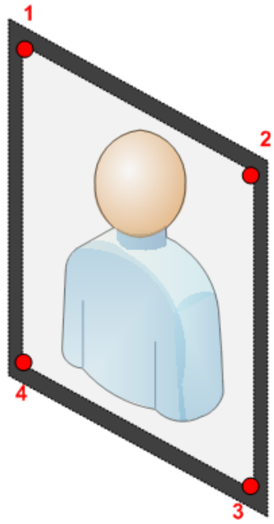

Figure 27 - Display position

4.3.2 The Multiview Video Information event package

The Multiview Video Information document defined in the previous section is shared between SIP user agents and application servers by means of the SIP event notification framework [RFC 3265]. Network entities will be able to query a user agent's multiview capabilities and topology by sending a SIP SUBSCRIBE request, and receive periodical SIP NOTIFY messages with the mvv-info in the body. A new SIP event package called 'mvv-info' has been defined for the subscription to multiview video information. The rest of the section explains this event package in detail.

4.3.2.1 Package Name

The name of the event package is "mvv-info". This value is used in the Event and Allow-Events, as defined in [RFC 3265]

4.3.2.2 SUBSCRIBE Bodies

SUBSCRIBE requests may contain a body. Typically, the purpose of a SUBSCRIBE body is to define a series of filters for the subscription. Filters for the "mvv-info" package are not specified in this document, but can be considered as a subject for future specifications. A SUBSCRIBE for this event package that is sent without a body implies a default filtering policy:

- The initial NOTIFY includes full state information.
- Each change in the MVV information of the notifier UA triggers a new NOTIFY message.
- Subsequent NOTIFY messages include a state delta (or full state information if partial notification is not supported).

4.3.2.3 Subscription Duration

The default expiration time for subscriptions to the "mvv-info" package is 3600 seconds. An alternate expiration may be defined in the Expires header of the SUBSCRIBE request.

4.3.2.4 NOTIFY Bodies

Notification bodies in this event package contain a MVV information document. These bodies are in a format listed in the Accept header field of the SUBSCRIBE request. By default, the format used for the body is "application/mvv-info+xml", as defined in 4.3.1. This format is used if no Accept header is present; all subscribers and notifiers must support it.

If the Accept header is present in the SUBSCRIBE request, it must include "application/mvv-info+xml", and may include other types.

4.3.2.5 Notifier Processing of SUBSCRIBE requests

A multiview video information document can include sensitive information. Because of this, subscriptions to this event package should be authenticated and subject to authorization.

4.3.2.6 Notifier generation of NOTIFY requests

This section provides information on the generation of NOTIFY messages, including when to send a NOTIFY, how to obtain MVV state information, and how to encrypt notification bodies.

4.3.2.6.1 Events causing a NOTIFY

Notifications should be sent in the following scenarios:

- As a response to a SUBSCRIBE request
- When the local MVV information changes. This can be due to a change in topology (e.g. a user, capture device, or display is moved, added, or removed) or in capabilities (such as a reduction in available bandwidth).

4.3.2.6.2 Computing state information for NOTIFY

The means by which a UA learns about its MVV information fall outside the scope of this document. Multiview capability information like available bandwidth and supported number of media streams may be defined statically in a configuration file, or be subject to change and continually monitored. Likewise, the topology of a conferencing site may be manually introduced and stored as a file, for very stable setups, or change frequently and require automated monitoring.

4.3.2.6.3 Encryption

Due to privacy concerns, it may be required to encrypt the notification bodies. A mechanism that can be used to provide integrity and authentication for these bodies is S/MIME. The usage of S/MIME with SIP is described in [RFC 3261].

4.3.2.7 Subscriber Processing of NOTIFY requests

A subscriber to this event package needs to keep track of the version number of the last mvv-info document received for a subscription (the value of the "version" attribute of "mvv-info"). When a new NOTIFY is received, if the new version number is equal or less than this number, the document must be discarded without processing. Otherwise, the subscriber should update the version number to the new version, and replace the local state with the new document.

The size of the MVV information documents can make it desirable to send updates with state deltas rather than the full state information. The definition of a mechanism for partial notification falls outside the scope of this document. However, future specifications can update the event package to support partial notifications.

4.3.2.8 Handling of forked requests

In this specification, forked SUBSCRIBE requests are not allowed to install multiple subscriptions. The notifier must guarantee that a single dialog is created in response to a SUBSCRIBE request, as described in section 4.4.9 of [RFC 3265]

4.3.2.9 Rate of notifications

A notifier should not generate notifications for a single subscriber at a rate of more than once every 5 seconds.

4.3.2.10 State Agents

The MVV information package is not particularly suited for the use of state agents. The topology and MVV capabilities of a conferencing site are naturally associated with a single

This package is not expected to benefit from the use of state agents as aggregation points or as nodes acting on behalf of other nodes. The MVV information of a conferencing site is naturally associated with a single user agent at that conferencing site, and that information will typically only be requested as a preliminary step to the initiation of a conference with that user agent.

4.3.2.11 Example Message Flow

This message flow illustrates how a user agent can send notifications for multiview video state to a subscriber. The user agent A (sip:a@example.com) subscribes to the multiview information of user agent B (sip:b@example.com), and receives an immediate notification containing its MVV information state. Later on, the MVV information of B changes, and B sends a new notification with the updated state.

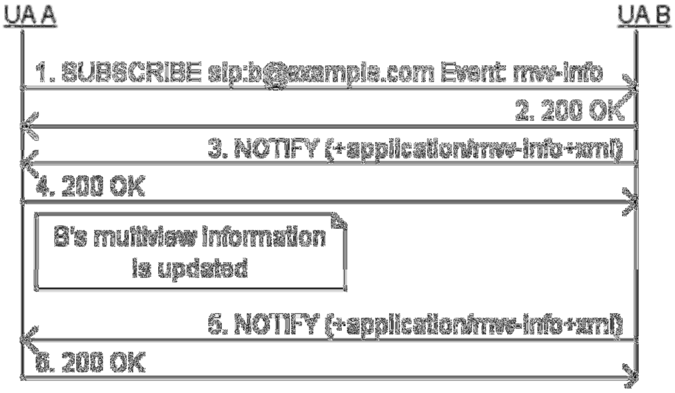
UA A
UA B
1. SUBSCRIBE sip:b@example.com Event: mvv-info
2. 200 OK
3. NOTIFY (+application/mvv-info+xml)
4. 200 OK
B's multiview information is updated
5. NOTIFY (+application/mvv-info+xml)
6. 200 OK

Figure 28 - Example Message Flow for mvv-info subscription

```
1. SUBSCRIBE   A->B

      SUBSCRIBE sip:b@example.com SIP/2.0
```

```
        Via: SIP/2.0/TCP a-host.example.com
        To: <sip:b@example.com>
        From: <sip:a@example.com>;tag=1111
        Call-ID: 1234@a-host.example.com
        CSeq: 1 SUBSCRIBE
        Max-Forwards: 70
        Event: mvv-info
        Accept: application/mvv-info+xml
        Contact: <sip:a@example.com>
        Expires: 3600
        Content-Length: 0

   2. 200 OK   B->A

        SIP/2.0 200 OK
        Via: SIP/2.0/TCP a-host.example.com
        To: <sip:b@example.com>;tag=2222
        From: <sip:a@example.com>;tag=1111
        Call-ID: 1234@a-host.example.com
        CSeq: 1 SUBSCRIBE
        Expires: 3600
        Contact: <sip:b@example.com>
        Content-Length: 0

    3. NOTIFY  B->A

        NOTIFY sip:a@example.com SIP/2.0
        Via: SIP/2.0/TCP b-host.example.com
        To: <sip:a@example.com>
        From: <sip:b@example.com>;tag=3333
        Call-ID: 1234@a-host.example.com
        Event: mvv-info
        Subscription-State: active;expires=3599
        Max-Forwards: 70
        CSeq: 1 NOTIFY
        Contact: sip:b.example.com
        Content-Type: application/mvv-info+xml
        Content-Length: ...

        [mvv-info Document]


     4. 200 OK A->B

        SIP/2.0 200 OK
        Via: SIP/2.0/TCP b-host.example.com
        To: <sip:a@example.com>;tag=4444
        From: <sip:b@example.com>;tag=3333
        Call-ID: 1234@a-host.example.com
        CSeq: 1 NOTIFY
        Content-Length: 0

    F5 NOTIFY  B->A
        NOTIFY sip:a@example.com SIP/2.0
        Via: SIP/2.0/TCP b-host.example.com
        To: <sip:a@example.com>
        From: <sip:b@example.com>;tag=3333
        Call-ID: 1234@a-host.example.com
        Event: mvv-info
        Subscription-State: active;expires=3500
        Max-Forwards: 70
        CSeq: 2 NOTIFY
        Contact: sip:b.example.com
        Content-Type: application/mvv-info+xml
        Content-Length: ...

        [mvv-info Document]


    6. 200 OK A->B

        SIP/2.0 200 OK
        Via: SIP/2.0/TCP b-host.example.com
        To: <sip:a@example.com>;tag=4444
        From: <sip:b@example.com>;tag=3333
        Call-ID: 1234@a-host.example.com
        CSeq: 2 NOTIFY
        Content-Length: 0
```

4.3.2.12 *Example mvv-info document*

In this section, we provide a mvv-info document for an example conferencing site. Consider a conferencing site M, with one user, two screens, and two cameras. The coordinates for these elements in the space of the site are listed in Table *16*.

Table 19 Element coordinates for example conferencing site M

Users					
Id	**position**				
u-m1	0,0,1200				
Displays					
Id	**P1**	**P2**	**P3**	**P4**	
d-m1	-1350,1620,850	-650,1840,850	-1350,1620,1300	-650,1840,1300	
d-m2	650,1840,850	1350,1620, 850	650,1840,1300	1350,1620 ,1300	
Captures					
Id	**Position**	**AoC-P1**	**AoC-P2**	**AoC-P3**	**AoC-P4**
c-m1	-1000,1730,1200	-1350,1620,850	-650,1840,850	-1350,1620,1300	-650,1840,1300
c-m2	1000,1730,1200	650,1840,850	1350,1620, 850	650,1840,1300	1350,1620 ,1300

The disposition of user, screens, and displays at the site are shown in Figure 29:

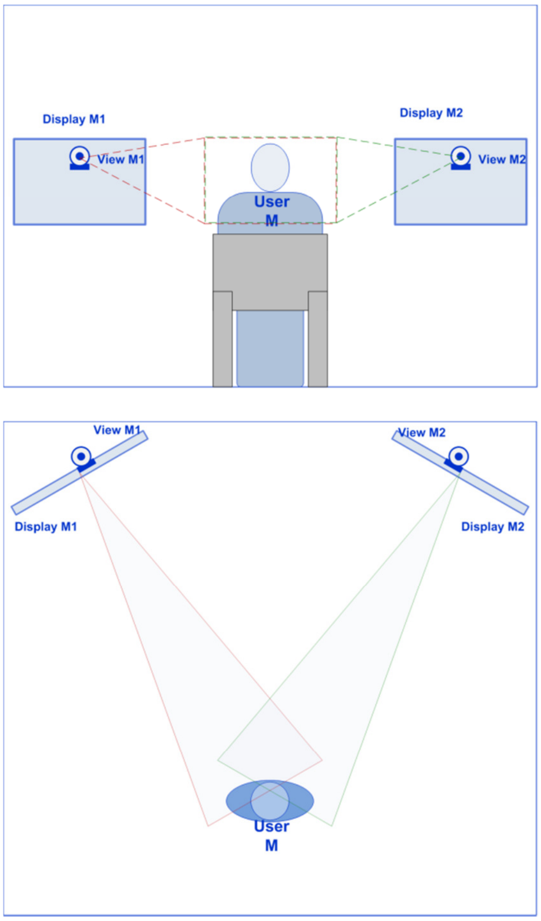

Figure 29 Example conferencing site, vertical view (top) and horizontal view (bottom)

The MVV information document for this site would be as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<mvv-info
xmlns="http://jungla.dit.upm.es/~capelastegui/mvv-info"
entity="sip:b@example.com"
version="1">
<!-- MVV CAPABILITIES-->
 <mvv-capabilities>
   <max-tx-bw>1000</max-tx-bw>
   <max-rx-bw>2000</max-rx-bw>
   <max-tx-captures media-type="video">2</max-tx-captures>
   <max-rx-captures media-type="video">4</max-rx-captures>
   <max-tx-captures>3</max-tx-captures>
   <max-rx-captures>6</max-rx-captures>
     <supported-formats>
      <encoding media=video name=H263-1998/>
      <encoding media=video name=H264/>
     <supported-formats/>
 </mvv-capabilities>
 <!--LIST OF USERS-->
 <user-list>
 <user id="u-m1">
   <position>
         <point x="0" y="0" z="1200"/>
      </position>
   <description>User at Madrid site</description>
 </user>
 </user-list>

 <!--LIST OF DISPLAYS-->
 <display-list>
 <display id="d-m1">
   <media-type>video</media-type>
   <position>
     <point x="-1350" y="1620" z="850"/>
     <point x="-650" y="1840" z="850"/>
     <point x="-1350" y="1620" z="1300"/>
     <point x="-650" y="1840" z="1300"/>
   </position>
   <associated-users><user id="u-m1"></associated-users>
 </display>
 <display id="d-m2">
   <media-type>video</media-type>
   <position>
     <point x="650" y="1840" z="850"/>
     <point x="1350" y="1620" z="850"/>
     <point x="650" y="1840" z="1300"/>
     <point x="1350" y="1620" z="1300"/>
   </position>
   <associated-users><user id="u-m1"></associated-users>
 </display>
 </display-list>

<!-- LIST OF CAPTURES -->
 <capture-list>
 <capture id="c-m1">
   <media-type>video</media-type>
   <position position-type="fixed">
         <point x="-500" y="1000" z="1500"/>
      </position>
   <capture-area>
     <point x="-1350" y="1620" z="850"/>
     <point x="-650" y="1840" z="850"/>
     <point x="-1350" y="1620" z="1300"/>
     <point x="-650" y="1840" z="1300"/>
   </capture-area>
   <max-bw>450</max-bw>
      <src-id>11111</src-id>
   <associated-users><user id="u-m1"></associated-users>
 </capture>
 <capture id="c-m2">
   <media-type>video</media-type>
   <position position-type="fixed">
         <point x="500" y="1000" z="1500"/>
      </position>
   <capture-area>
     <point x="650" y="1840" z="850"/>
```

```
      <point x="1350" y="1620" z="850"/>
      <point x="650" y="1840" z="1300"/>
      <point x="1350" y="1620" z="1300"/>
    </capture-area>
    <max-bw>450</max-bw>
    <src-id>22222</src-id>
    <associated-users><user id="u-m1"></associated-users>
  </capture>
  </capture-list>

</mvv-info>
```

4.3.3 Conferencing sites with multiview displays

The use of multiview displays in a conferencing site to show different video views to each user at the site (as explained in section 5.4) is a technique that can be used to improve eye contact in conferences with multiple users per site. When this kind of displays are used, they need to be reflected in the model included in a mvv-info document. The following rule applies:

A display that can render different views to each observing user is treated as **two separate <display> elements** in a MVV information document. Both <display> elements share the same media type and position, but have a different set of associated users.

This allows the conference configuration process to independently assign video views to each user observing the display.

4.4 Describing information of a MVV conference

The initiation of a conventional multimedia session using the SIP protocol normally involves configuring a series of media streams to be exchanged between two or more users. In the case of MVV conferences, a new layer is added to the process in the form of spatial information. One of the defining qualities of a MVV conference is the definition of a conference virtual space, as a tool to enable the provision of features like eye contact, spatial faithfulness, life-sized video, or free-viewpoint video. Also, complementing this virtual space, there is a map of media streams for the conference, describing for each stream which participants will receive it and how they will render it.

This virtual spaces and stream map need to be generated by a centralized entity compiling information from all conference participants, and be transmitted to user agents that will participate in the conference. However, this cannot be done as part of the offer/answer exchange of session descriptions typical of SIP session negotiation, since SDP documents are not suitable for conveying virtual space information, and user agents need this information as a prerequisite for the generation of the SDP offers and answers.

In this section, we propose a XML document format for describing conference virtual spaces and stream maps, and a mechanism for SIP entities to exchange this document.

The operation of this event package is summarized as follows:

1) A conference focus is asked to initiate a MVV conference for a group of participants, whose multiview video information (see 4.3) is known
2) The focus generates a model of the conference, including one or more conference virtual spaces and a map of media streams
3) For each participant, the focus generates a MVV event document based on the conference model, and sends it to the participant UA.
4) When all participants have received the MVV events, the conference can be initiated.

4.4.1 MVV Conference Information Document

The multiview video conference information document (“mvv-conf-info”) is an XML document associated with a SIP conference (as defined in [rfc4353]) in which multiview video functionality is used. It includes auxiliary information that is used in the configuration of the conference, including the definition of one or more virtual spaces describing the spatial relationships between conference elements, and a stream map listing and describing the media streams exchanged in the conference.

The following diagram gives an example of the structure used by the mvv-conf-info. The document includes a series of virtual spaces containing users and displays, and a map of all the streams exchanged in the conference. Document elements that can appear in multiples are marked with a ‘*’, and optional elements are enclosed in square brackets: ‘[ ]’.

```
mvv-conf-info
|
|-- virtual-space (entity, [user]) *
| |-- user-list
| |   |-- user (id, entity)*
| |   | |-- position
| |   | |-- [description]
| |-- display-list
| |   |-- display (id, entity)*
| |   | |-- media-type
| |   | |-- position
| |   | |-- capture *
| |-- capture-list
| |   |-- capture (id, entity)*
| |   | |-- media-type
| |   | |-- label
| |   | |-- position
| |   | |-- capture-area
|
|-- stream-map
| |-- supported-formats
| |   |-- encoding (media-type, name)
| |-- endpoint (entity)*
| |   |-- capture (id)*
| |   | |-- media-type
| |   |   |-- label
| |   | |-- associated-users
| |   | | |-- user (id)
| |   | |-- receivers
| |   | | |-- receiver (entity)*
| |   | |-- [max-bw]
| |   | |-- [src-id]
| |   | |-- [position-stream-label]
| |   | |-- [control-stream-label]
| |   |-- auxiliary-stream (id)*
| |   | |-- media-type
| |   | |-- auxiliary-function
| |   | |-- label
| |   | |-- receivers
| |   | | |-- receiver (entity)*
| |   | |-- [max-bw]
| |   | |-- [src-id]
```

The elements of the document are described in the following subsections

4.4.1.1 <mvv-conf-info>

The multiview video conference information document has the root element <mvv-conf-info>. The following mandatory attributes have been defined for <mvv-conf-info >: “**entity**” and “**version**”. In addition, mvv-info elements have an associated XML namespace name: http://jungla.dit.upm.es/~capelastegui/mvv-conf-info . This namespace is declared using an “xmlns” attribute, as described in [XML-NS], and can be used as a default namespace or associated with a namespace prefix.

- **entity**: This attribute contains the conference URI identifying the conference associated with the document. This SIP URI is used for subscriptions to this conference, and may also be used for invitations to the conference.
- **version**: This attribute is an integer that increases by one between subsequent document updates. This allows the use of “state delta” processing of multiview information documents, as described in [RFC3265].

4.4.1.2 <virtual-space>

This element describes the topology of a virtual space for the MVV conference. Users and display devices of all conference participants are represented in this virtual space. The virtual space has the following attributes: “**entity**”, and “**user**”.

- **entity**: A mandatory attribute containing a SIP URI identifying the user agent associated with the virtual space. The value of “entity” may coincide with that of the “entity” attribute of the root <mvv-conf-info> element; if that is the case, the virtual space is considered the common virtual space for the conference. Any user agent without a defined virtual space must use the common virtual space. It is not mandatory for a conference to have a common virtual space, but if it lacks one, every user agent in the conference must have a defined virtual space.
- **user:** An optional attribute containing the identifier of a user in the conference. This identifier must match that of one <user> element associated with the user agent corresponding to this virtual space. This attribute is only used when a single virtual space cannot be used for all users at a conferencing site, which can happen in scenarios with multiview displays (see 5.4), for example.

The virtual space has two child elements, <user-list> and <display-list>, defined in the following sections. This element can be extended in the future to include other objects within the virtual space.

Note that some child elements of the virtual space include references to positions in a coordinate system. If possible, the coordinate system of a conference virtual space for a given user agent should have the same origin and orientation as that of the coordinate system used in the MVV information document for that user agent (see 4.3). The coordinate system for the common conference virtual space can have an arbitrary origin.

4.4.1.3 <user-list> and <user> sub-elements

The <**user-list**> element includes a series of <user> child elements, each representing a user in the conference virtual space.

Each <**user**> element has two mandatory attributes, “**id**”, and “**entity**”. The “id” attribute contains a unique identifier in the context of the terminal of this user. This attribute corresponds to the “id” attribute of the <user> element for this user in the MVV information document (see 4.3.1.4) of the user's terminal. The "entity" attribute contains the SIP URI of the user's terminal.

Each user has the following child elements:

- <**position**>: The position of the user. It includes one <point> element with three attributes 'x', 'y', and 'z' representing coordinates, in millimetres, within the Cartesian coordinate system defined for the conference virtual space. This element corresponds to the <position> element for this user in the MVV information document (see 4.3.1.4) of the user's terminal.
- <**description**>: Optional element with a textual description of the user. This will typically include the user's name or role.

4.4.1.4 <display-list> and <display> sub-elements

The optional element <**display-list**> represents the list of displays in the virtual space. It includes one or more <**display**> elements.

Each <**display**> has two mandatory attributes, "**id**", and "**entity**". The "id" attribute contains a unique identifier in the context of the terminal of this display. This attribute corresponds to the "id" attribute of the <display> element for this display device in the MVV information document (see 4.3.1) of the terminal. The "entity" attribute contains the SIP URI of the terminal.

A <display> can have the following mandatory child elements:

- <**media-type**>: The media type displayed by this device. Must be one media type registered for "media" in SDP [RFC 4566], like "audio" or "video".
- <**position**>: The position of the display. It includes four <point> element, each with three attributes 'x', 'y', and 'z' representing coordinates, in millimetres. The four <point> elements correspond to the corners of the viewable area of the display, and should be coplanar and form a quadrilateral.
- <**capture>**: A media stream rendered by the display. It contains a identifier for the stream. The value of this identifier must match that of the <label> element within a <capture> element in the stream map of this document. A display can contain multiple <capture> elements.

4.4.1.5 <capture-list> and <capture> sub-elements

The optional element <**capture-list**> represents the list of capture devices in the virtual space. It includes one or more <**capture**> elements.

Each <**capture**> has two mandatory attributes, "**id**", and "**entity**". The "id" attribute contains a unique identifier in the context of the terminal of this display. This attribute corresponds to the "id" attribute of the <capture> element for the media stream captured by this device in the MVV information document (see 4.3.1) of the terminal. The "entity" attribute contains the SIP URI of the terminal.

A <capture> can have the following mandatory child elements:

- <**media-type**>: The media type displayed by this device. Must be one media type registered for "media" in SDP [RFC 4566], like "audio" or "video".
- <**label**>: A unique identifier for the captured stream in the context of the conference. The value of this element corresponds to the "label" attribute in SDP, defined in [RFC 4574].
- <**position**>: It has an optional 'position-type' attribute, which can have two possible values: "fixed" and "dynamic". "fixed" is the default value, used when the capture position is a fixed point in space. Captures with "dynamic" position, on the other hand, can be located in an arbitrary position within a certain range, which can change dynamically over the course of a conference. The value of 'position-type' matches its analogue in the mvv-info document (see 4.3.1) of the user agent associated with this stream, with one exception: streams which originally had a position-type of "variable" are configured with a "fixed"position in this document. A <position> can have the following child elements:
  - <**point**>: An element with three attributes 'x', 'y', and 'z' representing coordinates, in millimetres, within the Cartesian coordinate system defined for the conference virtual space. This element is optional if the stream has a "position-type" value of "dynamic"; in that scenario, the point represents the initial capture position at the start of the conference.
  - <**position-range>:** An element representing the range of possible capture positions, when the position type is dynamic. It contains several <point> elements defining the boundaries of the position range. A <position-range> can have 2 <point> elements (defining a line), 4 <point> elements (defining a quadrilateral), or 8 <point> elements (defining an hexahedron). This element must be omitted if the <position> has a fixed position type, otherwise it is mandatory. It is possible to have multiple <position-ranges> within a <position> element. In this case, all points included in these <position-ranges> are valid capture positions.
- <**capture-area**> : The area of space captured in the stream. It includes four <point> elements, each with three attributes 'x', 'y', and 'z' representing coordinates in millimetres. The four <point> elements should be coplanar and form a quadrilateral. Alternately, when the <position> element of this capture has a "position-type" value of "dynamic", these point elements can be replaced by a <capture-range> element:
  - <**capture-range>:** An element representing the range of possible positions for a capture area, when the stream has a dynamic position type. A <capture-range> can have 8 <point> elements (defining an hexahedron). The actual capture area is defined in an auxiliary data stream – de details of which are defined in the stream map of a mvv-conf-info document. It is possible to have multiple <capture-ranges> within a <capture> element. In this case, all points included in these <capture-ranges> are valid capture area positions – however, all points in a capture area need to fit within a single <capture-range>

4.4.1.6 <stream-map> and <endpoint> child elements

The <**stream-map**> element represents a list of all the media streams exchanged in the conference. It includes a series of <**endpoint**> child elements, each representing a user agent participating in the conference. Each <endpoint> has a mandatory "**entity**" attribute containing the SIP URI of that device, and a series <**capture**> child elements, which are defined in the following section. Optionally, an endpoint can also have a <**supported-formats**> child element. In conferences with free-viewpoint-video, endpoints can have additional elements, <auxiliary-stream>, representing auxiliary media streams associated with a free-viewpoint-video stream.

4.4.1.7 <capture>

Each <**capture**> represents a media stream transmitted in this conference by the user agent corresponding to its root <**endpoint>** element**.** It has a mandatory attribute, "**id**", containing a unique identifier in the context of the terminal capturing this stream. This attribute corresponds to the "id" attribute of the <capture> element for this stream in the MVV information document (see 4.3.1) of the capturing terminal.

A <**capture**> can have the following child elements:

- <**media**-**type**>: Media type for the stream.
- <**label**>: A unique identifier for the stream in the context of the conference. The value of this element corresponds to the "label" attribute in SDP, defined in [RFC 4574]..
- <**max-bw**>: Optional element representing the maximum bandwidth, in kilobits per second, of this media stream. This corresponds to the value of the 'b=' field in SDP [RFC 4566] , when the bandwidth type for that field is 'AS' (application specific).
- <**src-id>:** The identifier for the media source for this stream. For a RTP [RFC 3550] stream, this corresponds to the SSRC value.
- <**associated-users**>: An optional element listing the users captured in this media stream. If absent, all users at the capture site are assumed to be included in the stream. It includes one or more <user> elements with an 'id' attribute containing a unique user identifier. A user defined in a virtual space in this document is considered to match this <user> element if their 'id' attributes match, and the 'entity' attribute of the virtual space user matches that of the root <endpoint> element of this user.
- <**receivers**>: List of user agents participating in the conference that will receive this stream. It includes a series of <**receiver>** elements representing receiving user agents, each with an "entity" attribute containing the user agent SIP URI. If the stream is a free-viewpoint-video stream (i.e. it has been defined with a "position-type" of dynamic in the mvv-info document of its corresponding user agent), a <receiver> can have the following additional element:
    - <**role**>: Defines any special role performed by the receiving user agent regarding this stream. By default, this has a value of "receiver", for receiving user agents with no associated special functions. In the context of a free-viewpoint-video stream or other media stream with a dynamic capture position, this can have a value of "controller", to indicate that the receiving user agent is in charge of controlling the capture position.

- <**position-stream-label**>: Used only with media streams with dynamic capture position (such as free-viewpoint-video), this element contains a unique identifier for an auxiliary media stream associated with this media stream. The purpose of this auxiliary stream, which is sent by the same user agent as the media stream, is to provide real-time information regarding the capture position of the media stream.
- <**control-stream-label**> Used only with media streams with dynamic capture position (such as free-viewpoint-video), this element contains a unique identifier for an auxiliary media stream associated with this media stream. The purpose of this auxiliary stream is to allow a remote user agent to change the capture position of the media stream. The auxiliary stream is received by the sender of the dynamic media stream, and sent by a user agent identified as a receiver and a controller for that stream (i.e. included in the <receivers> element, and identified by a <role> child element with a value of "controller").

4.4.1.8 <supported-formats>

A <**supported-formats**> optional element represents a list of encoding names (e.g. RTP payload types) supported by the terminals participating in a conference. It includes one or more <**encoding**> elements with a **'media-type**' attribute containing the media type of the encoding (usually 'audio' or 'video) and a 'name' attribute containing the encoding name. It can also have an optional attribute '**fmtp'** including a string with format-specific parameters used for the encoding (as would be included in a 'a=fmtp:' attribute in SDP).

4.4.1.9 <auxiliary-stream>

A <auxiliary-stream> element represents an auxiliary stream for the transmission of positioning or control information associated with a media stream, which can be a free-viewpoint-video stream or another stream with a dynamic capture position. It can have the following child elements:

- <**media**-**type**>: Media type for the stream. These streams usually have the "application" media type.
- <**auxiliary**-**function** >: The function performed by the stream. Two values have been defined: "**position-stream"** and **"control-stream",** for streams transmitting real-time position information for a media stream, or commanding a capturing user agent to change the capture position of a media stream, respectively.
- <**label**>: A unique identifier for the stream in the context of the conference. The value of this element corresponds to the "label" attribute in SDP, defined in [RFC 4574].
- <**associated-stream-label**>: The label of the media stream for which this stream provides positioning information or control.
- <**max-bw**>: Optional element representing the maximum bandwidth, in kilobits per second, of this media stream. This corresponds to the value of the 'b=' field in SDP [RFC 4566] , when the bandwidth type for that field is 'AS' (application specific).
- <**src-id>:** The identifier for the media source for this stream. For a RTP [RFC 3550] stream, this corresponds to the SSRC value.

- **<receivers>**: List of user agents participating in the conference that will receive this stream. It includes a series of **<receiver>** elements representing receiving user agents, each with an "entity" attribute containing the user agent SIP URI.

4.4.2 The MVV Conference Information Event Package

The Multiview Video Conference Information document defined in the previous section is shared between SIP user agents and the conference focus by means of the SIP event notification framework [RFC 3265].

Once subscribed, user agents in a conference will be able to query the conference focus for updates on conference virtual spaces and stream maps by sending a SIP SUBSCRIBE request, and receive periodical SIP NOTIFY messages with the mvv-conf-info in the body. A new SIP event package called 'mvv-conf-info' has been defined for the subscription to multiview video information. The rest of the section explains this event package in detail.

4.4.2.1 Package Name

The name of the event package is "mvv-conf-info". This value is used in the Event and Allow-Events, as defined in [RFC 3265]

4.4.2.2 SUBSCRIBE Bodies

SUBSCRIBE requests may contain a body. Typically, the purpose of a SUBSCRIBE body is to define a series of filters for the subscription. Filters for the "mvv-conf-info" package are not specified in this document, but can be considered as a subject for future specifications. A SUBSCRIBE for this event package that is sent without a body implies a default filtering policy:

- The initial NOTIFY includes full state information.
- Each change in the MVV conference state triggers a new NOTIFY message.
- Subsequent NOTIFY messages include a state delta (or full state information if partial notification is not supported).

4.4.2.3 Subscription Duration

The default expiration time for subscriptions to the "mvv-conf-info" package is 3600 seconds. An alternate expiration may be defined in the Expires header of the SUBSCRIBE request.

4.4.2.4 NOTIFY Bodies

Notification bodies in this event package contain a MVV conference information document. These bodies are in a format listed in the Accept header field of the SUBSCRIBE request. By default, the format used for the body is "application/mvv-conf-info+xml", as defined in 4.4.14.3.1. This format is used if no Accept header is present; all subscribers and notifiers must support it.

If the Accept header is present in the SUBSCRIBE request, it must include "application/mvv-conf-info+xml", and may include other types.

4.4.2.5 Notifier Processing of SUBSCRIBE requests

A multiview video conference information document can include sensitive information. Because of this, subscriptions to this event package should be authenticated and subject to authorization.

4.4.2.6 Notifier generation of NOTIFY requests

This section provides information on the generation of NOTIFY messages, including when to send a NOTIFY, how to obtain MVV state information, and how to encrypt notification bodies.

4.4.2.6.1 Events causing a NOTIFY

Notifications should be sent in the following scenarios:

- As a response to a SUBSCRIBE request
- When the conference MVV information changes. This can be due to participants joining or leaving the conference, or updating their MVV information. It can also be motivated by a change in available network resources, or by the conference focus deciding to change conference configuration.

4.4.2.6.2 Encryption

Due to privacy concerns, it may be required to encrypt the notification bodies. A mechanism that can be used to provide integrity and authentication for these bodies is S/MIME. The usage of S/MIME with SIP is described in [RFC 3261].

4.4.2.7 Subscriber Processing of NOTIFY requests

A subscriber to this event package needs to keep track of the version number of the last mvv-info document received for a subscription (the value of the "version" attribute of "mvv-info"). When a new NOTIFY is received, if the new version number is equal or less than this number, the document must be discarded without processing. Otherwise, the subscriber should update the version number to the new version, and replace the local state with the new document.

The size of the MVV conference information documents can make it desirable to send updates with state deltas rather than the full state information. The definition of a mechanism for partial notification falls outside the scope of this document. However, future specifications can update the event package to support partial notifications.

4.4.2.8 Handling of forked requests

In this specification, forked SUBSCRIBE requests are not allowed to install multiple subscriptions. The notifier must guarantee that a single dialog is created in response to a SUBSCRIBE request, as described in section 4.4.9 of [RFC 3265]

4.4.2.9 Rate of notifications

A notifier should not generate notifications for a single subscriber at a rate of more than once every 5 seconds.

4.4.2.10 State Agents

The use of state agents as aggregation points is not considered for this event package. A possible scenario where state aggregation could be applied is that of a chained conference (see [RFC4353]) with MVV features. However, support for chained MVV conferences is not covered by this specification, and is left for future extensions.

4.4.2.11 Example Message Flow

This message flow illustrates how a user agent can send notifications for multiview video state to a subscriber. The user agent A (sip:a@example.com) subscribes to the multiview information of user agent B (sip:b@example.com), and receives an immediate notification containing its MVV information state. Later on, the MVV information of B changes, and B sends a new notification with the updated state.

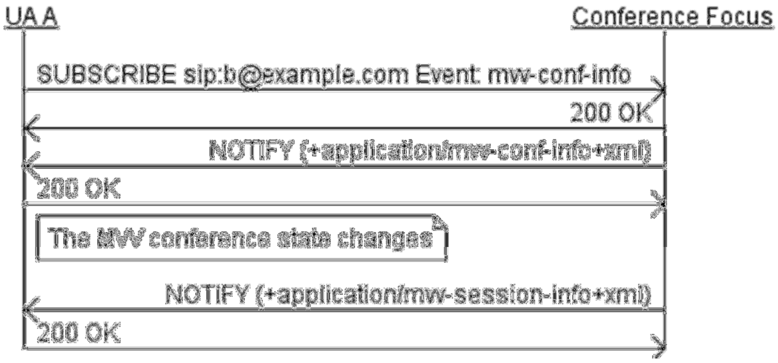
UA A
Conference Focus
SUBSCRIBE sip:b@example.com Event: mvv-conf-info
200 OK
NOTIFY (+application/mvv-conf-info+xml)
200 OK
The MVV conference state changes
NOTIFY (+application/mvv-session-info+xml)
200 OK

Figure 30 - Example Message Flow for mvv-info subscription

```
1. SUBSCRIBE   A->B

      SUBSCRIBE sip:b@example.com SIP/2.0
      Via: SIP/2.0/TCP a-host.example.com
      To: <sip:b@example.com>
      From: <sip:a@example.com>;tag=1111
      Call-ID: 1234@a-host.example.com
      CSeq: 1 SUBSCRIBE
      Max-Forwards: 70
      Event: mvv-conf-info
      Accept: application/mvv-conf-info+xml
      Contact: <sip:a@example.com>
      Expires: 3600
      Content-Length: 0

 2. 200 OK   B->A

      SIP/2.0 200 OK
      Via: SIP/2.0/TCP a-host.example.com
      To: <sip:b@example.com>;tag=2222
      From: <sip:a@example.com>;tag=1111
      Call-ID: 1234@a-host.example.com
      CSeq: 1 SUBSCRIBE
      Expires: 3600
      Contact: <sip:b@example.com>
      Content-Length: 0
```

```
3. NOTIFY  B->A

    NOTIFY sip:a@example.com SIP/2.0
    Via: SIP/2.0/TCP b-host.example.com
    To: <sip:a@example.com>
    From: <sip:b@example.com>;tag=3333
    Call-ID: 1234@a-host.example.com
    Event: mvv-conf-info
    Subscription-State: active;expires=3599
    Max-Forwards: 70
    CSeq: 1 NOTIFY
    Contact: sip:b.example.com
    Content-Type: application/mvv-conf-info+xml
    Content-Length: ...

    [mvv-info Document]


 4. 200 OK A->B

    SIP/2.0 200 OK
    Via: SIP/2.0/TCP b-host.example.com
    To: <sip:a@example.com>;tag=4444
    From: <sip:b@example.com>;tag=3333
    Call-ID: 1234@a-host.example.com
    CSeq: 1 NOTIFY
    Content-Length: 0

F5 NOTIFY  B->A

    NOTIFY sip:a@example.com SIP/2.0
    Via: SIP/2.0/TCP b-host.example.com
    To: <sip:a@example.com>
    From: <sip:b@example.com>;tag=3333
    Call-ID: 1234@a-host.example.com
    Event: mvv-info
    Subscription-State: active;expires=3500
    Max-Forwards: 70
    CSeq: 2 NOTIFY
    Contact: sip:b.example.com
    Content-Type: application/mvv-conf-info+xml
    Content-Length: ...

    [mvv-conf-info Document]


6. 200 OK A->B

    SIP/2.0 200 OK
    Via: SIP/2.0/TCP b-host.example.com
    To: <sip:a@example.com>;tag=4444
    From: <sip:b@example.com>;tag=3333
    Call-ID: 1234@a-host.example.com
```

```
     CSeq: 2 NOTIFY
  Content-Length: 0
```

4.4.3 Example MVV conference event

To illustrate how the extended conference event document works, we have defined an example MVV conference in detail, based on the conference described in section 4.1.1. For clarity's sake, the audio streams and devices (microphones, speakers) have been omitted from this example, and only video-related elements are shown.

Consider a video conference between three sites M (at Madrid), B (at Barcelona) and S (at Sevilla). Each conferencing site has one user, two displays, and two capture devices. The coordinates for these elements in the conference virtual space are shown in Table 20, Table 21, and Table 22.

Table 20 Example conference virtual space - users

Id	Entity	position
u-m1	sip:m@example.com	0,0,1200
u-b1	sip:b@example.com	-1000,1730,1200
u-s1	sip:s@example.com	1000,1730,1200

Table 21 Example conference virtual space - displays

Id	Entity	P1	P2	P3	P4
d-m1	sip:m@example.com	-1350,1620,850	-650,1840,850	-1350,1620,1300	-650,1840,1300
d-m2	sip:m@example.com	650,1840,850	1350,1620, 850	650,1840,1300	1350,1620 ,1300
d-b1	sip:b@example.com	1000,2130,850	1000, 1330,850	1000,2130,1300	1000,1330,1300
d-b2	sip:b@example.com	-350,-200,850	350,200,850	-350,-200,1300	350,200,1300
d-s1	sip:s@example.com	-350,200,850	350,-200,850	-350,200,1300	350,-200,1300
d-s2	sip:s@example.com	-1000,1330,850	-1000, 2130,850	-1000,1330,1300	-1000,2130,1300

Table 22 Example conference virtual space - captures

Entity: sip:m@example.com					
Label	**Position**	**AoC-P1**	**AoC-P2**	**AoC-P3**	**AoC-P4**
c-m-b	-1000,1730,1200	-1350,1620,850	-650,1840,850	-1350,1620,1300	-650,1840,1300
c-m-s	1000,1730,1200	650,1840,850	1350,1620, 850	650,1840,1300	1350,1620 ,1300
Entity: sip:b@example.com					
Label	**Position**	**AoC-P1**	**AoC-P2**	**AoC-P3**	**AoC-P4**
c-b-s	1000,1730,1200	1000,2130,850	1000, 1330,850	1000,2130,1300	1000,1330,1300
c-b-m	0,0,1200	-350,-200,850	350,200,850	-350,-200,1300	350,200,1300
Entity: sip:s@example.com					
Label	**Position**	**AoC-P1**	**AoC-P2**	**AoC-P3**	**AoC-P4**
c-s-m	0,0,1200	-350,200,850	350,-200,850	-350,200,1300	350,-200,1300
c-s-b	-1000,1730,1200	-1000,1330,850	-1000, 2130,850	-1000,1330,1300	-1000,2130,1300

The disposition of users, screens, and capture devices at the conference virtual space are shown in Figure 31.

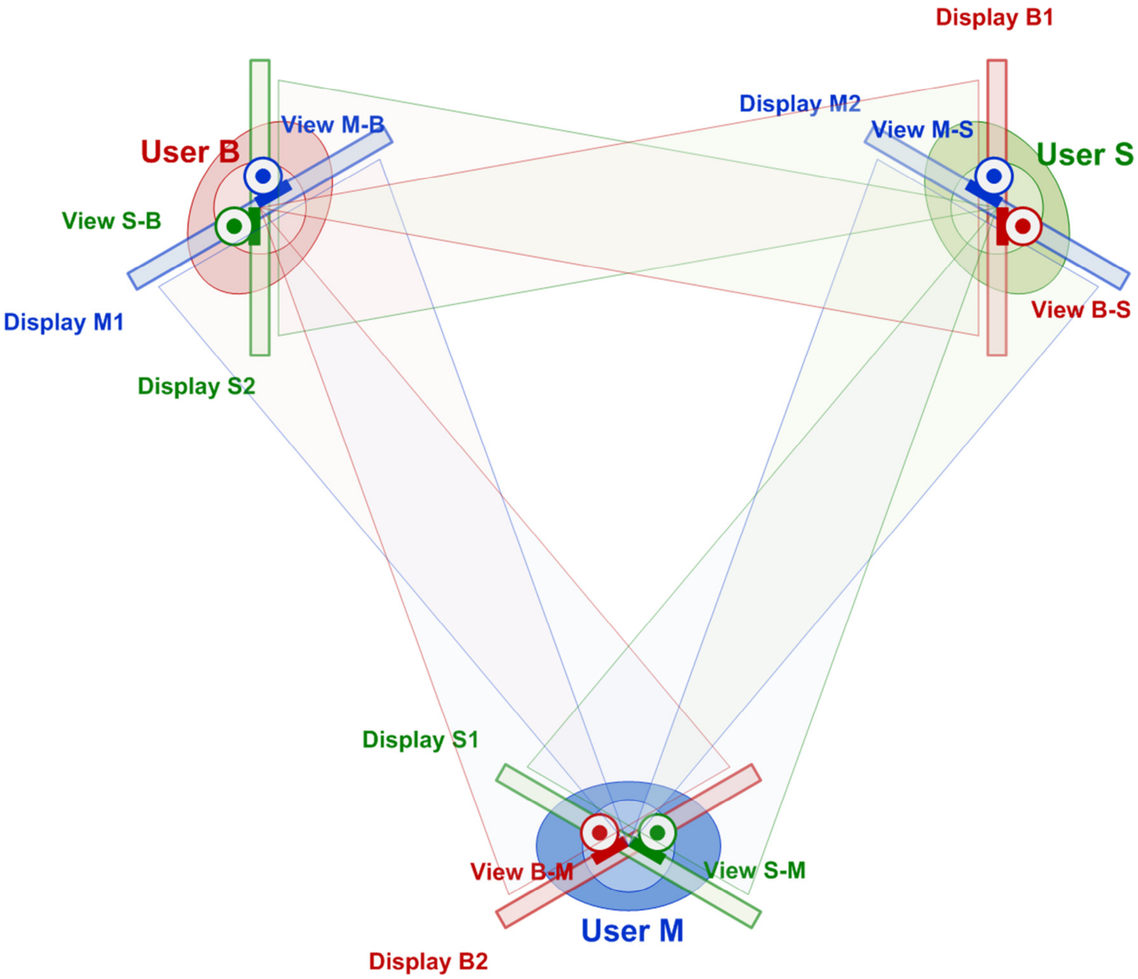

Figure 31 Example conference virtual space

The MVV conference information document for this conference would be as follows. The conference virtual space included in the document is common to all users.

```
<?xml version="1.0" encoding="UTF-8"?>
<mvv-conf-info xmlns="http://jungla.dit.upm.es/~capelastegui/mvv-conf-
info" entity="sip:conf1@example.com" version="1">
   <!-- VIRTUAL SPACE USER M -->
<virtual-space entity="sip:conf1@example.com">
   <!-- LIST OF USERS -->

 <user-list>
  <user id="u-m1" entity="sip:m@example.com">
   <position>
     <point x="0" y="0" z="1200"/>
   </position>
   <description>Madrid user 1</description>
  </user>
  <user id="u-b1" entity="sip:b@example.com">
    <position>
     <point x="-1000" y="1730" z="1200"/>
   </position>
   <description>Barcelona user 1</description>
  </user>
  <user id="u-s1" entity="sip:s@example.com">
   <position>
    <point x="1000" y="1730" z="1200"/>
  </position>
   <description>Sevilla user 1</description>
  </user>
 </user-list>

   <!-- LIST OF DISPLAYS -->


 <display-list>
  <display id="d-m1" entity="sip:m@example.com">
   <media-type>video</media-type>
   <position>
    <point x="-1350" y="1620" z="850"/>
    <point x="-650" y="1840" z="850"/>
  <point x="-1350" y="1620" z="1300"/>
    <point x="-650" y="1840" z="1300"/>
   </position>
   <capture>v-B-M</capture>
  </display>
  <display id="d-m2" entity="sip:m@example.com">
   <media-type>video</media-type>
   <position>
    <point x="650" y="1840" z="850"/>
    <point x="1350" y="1620" z="850"/>
    <point x="650" y="1840" z="1300"/>
  <point x="1350" y="1620" z="1300"/>
   </position>
   <capture>v-S-M</capture>
  </display>

  <display id="d-b1" entity="sip:b@example.com">
   <media-type>video</media-type>
   <position>
    <point x="1000" y="2130" z="850"/>
    <point x="1000" y="1330" z="850"/>
  <point x="1000" y="2130" z="1300"/>
    <point x="1000" y="1330" z="1300"/>
   </position>
   <capture>v-S-B</capture>
  </display>
  <display id="d-b2" entity="sip:b@example.com">
   <media-type>video</media-type>
   <position>
    <point x="-350" y="-200" z="850"/>
    <point x="350" y="200" z="850"/>
    <point x="-350" y="-200" z="1300"/>
  <point x="350" y="200" z="1300"/>
   </position>
```

```
   <capture>v-M-B</capture>
  </display>

  <display id="d-s1" entity="sip:s@example.com">
   <media-type>video</media-type>
   <position>
    <point x="-350" y="200" z="850"/>
    <point x="350" y="-200" z="850"/>
    <point x="-350" y="200" z="1300"/>
  <point x="350" y="-200" z="1300"/>
   </position>
   <capture>v-M-S</capture>
  </display>
  <display id="d-s2" entity="sip:s@example.com">
   <media-type>video</media-type>
   <position>
    <point x="-1000" y="1330" z="850"/>
    <point x="-1000" y="2130" z="850"/>
    <point x="-1000" y="1330" z="1300"/>
  <point x="-1000" y="2130" z="1300"/>
   </position>
   <capture>v-B-S</capture>
  </display>
 </display-list>

<!-- LIST OF CAPTURES -->

 <capture-list>
  <capture id="c-m1" entity="sip:m@example.com">
   <media-type>video</media-type>
   <label>c-M-B</label>
   <position>
     <point x="-1000" y="1730" z="1200"/>
   </position>
   <capture-area>
    <point x="-350" y="-200" z="850"/>
    <point x="350" y="200" z="850"/>
    <point x="-350" y="-200" z="1300"/>
  <point x="350" y="200" z="1300"/>
   </capture-area>
  </capture>
  <capture id="c-m2" entity="sip:m@example.com">

   <media-type>video</media-type>
   <label>c-M-S</label>
   <position>
    <point x="1000" y="1730" z="1200"/>
   </position>
   <capture-area>
    <point x="-350" y="200" z="850"/>
    <point x="350" y="-200" z="850"/>
    <point x="-350" y="200" z="1300"/>
  <point x="350" y="-200" z="1300"/>
 </capture-area>
</capture>
<capture id="c-b1" entity="sip:b@example.com">
 <media-type>video</media-type>
 <label>c-B-S</label>
 <position>
  <point x="1000" y="1730" z="1200"/>
 </position>
 <capture-area>
  <point x="-1000" y="1330" z="850"/>
  <point x="-1000" y="2130" z="850"/>
  <point x="-1000" y="1330" z="1300"/>
<point x="-1000" y="2130" z="1300"/>
 </capture-area>
</capture>
<capture id="c-b2" entity="sip:b@example.com">
 <media-type>video</media-type>
 <label>c-B-M</label>
 <position>
   <point x="0" y="0" z="1200"/>
 </position>
 <capture-area>
  <point x="-1350" y="1620" z="850"/>
  <point x="-650" y="1840" z="850"/>
<point x="-1350" y="1620" z="1300"/>
  <point x="-650" y="1840" z="1300"/>
 </capture-area>
</capture>

<capture id="c-s1" entity="sip:s@example.com">
 <media-type>video</media-type>
 <media-type>video</media-type>
 <label>c-S-M</label>
 <position>
   <point x="0" y="0" z="1200"/>
 </position>
 <capture-area>
  <point x="650" y="1840" z="850"/>
  <point x="1350" y="1620" z="850"/>
  <point x="650" y="1840" z="1300"/>
<point x="1350" y="1620" z="1300"/>
 </capture-area>
</capture>
<capture id="c-s2" entity="sip:s@example.com">
 <media-type>video</media-type>
 <label>c-S-B</label>
 <position>
   <point x="-1000" y="1730" z="1200"/>
 </position>
 <capture-area>
  <point x="1000" y="2130" z="850"/>
  <point x="1000" y="1330" z="850"/>
<point x="1000" y="2130" z="1300"/>
```

```
    <point x="1000" y="1330" z="1300"/>
   </capture-area>
  </capture>
 </capture-list>

</virtual-space>

   <!-- MAP OF STREAMS -->

<stream-map>
  <endpoint entity="sip:m@example.com">
    <supported-formats>
      <encoding media-type=video name=H263-1998/>
      <encoding media-type=video name=H264/>
    <supported-formats/>
   <capture id="c-m1">
    <media-type>video</media-type>
    <label>c-M-B</label>
  <associated-users>
   <user id="u-m1"/>
  </associated-users>
  <receivers>
   <receiver entity="sip:b@example.com"/>
  </receivers>
  <max-bw>450</max-bw>
  <src-id>11111 </src-id>

   </capture>
      <capture id="c-m2">
    <media-type>video</media-type>
    <label>c-M-S</label>
  <associated-users>
   <user id="u-m1"/>
  </associated-users>
  <receivers>
   <receiver entity="sip:s@example.com"/>
  </receivers>
  <max-bw>450</max-bw>
  <src-id>22222 </src-id>
   </capture>
  </endpoint>

  <endpoint entity="sip:b@example.com">
   <supported-formats>
      <encoding media-type=video name=H263-1998/>
      <encoding media-type=video name=H264/>
    <supported-formats/>
   <capture id="c-b1">
    <media-type>video</media-type>
    <label>c-B-S</label>
  <associated-users>
   <user id="u-b1"/>
  </associated-users>
```

```
<receivers>
 <receiver entity="sip:s@example.com"/>
</receivers>
<max-bw>450</max-bw>
<src-id>3333 </src-id>
 </capture>

 <capture id="c-b2">
  <media-type>video</media-type>
  <label>c-B-M</label>
<associated-users>
 <user id="u-b1"/>
</associated-users>
<receivers>
 <receiver entity="sip:m@example.com"/>
</receivers>
<max-bw>450</max-bw>
<src-id>44444 </src-id>
 </capture>
</endpoint>

<endpoint entity="sip:s@example.com">
 <supported-formats>
     <encoding media-type=video name=H263-1998/>
    <encoding media-type=video name=H264/>
 <supported-formats/>
 <capture id="c-s1">
  <media-type>video</media-type>
  <label>c-S-M</label>
<associated-users>
 <user id="u-s1"/>
</associated-users>
<receivers>
 <receiver entity="sip:m@example.com"/>
</receivers>
<max-bw>450</max-bw>
<src-id>55555 </src-id>
 </capture>

 <capture id="c-s2">
  <media-type>video</media-type>
  <label>c-S-S</label>
<associated-users>
 <user id="u-s1"/>
</associated-users>
<receivers>
 <receiver entity="sip:b@example.com"/>
</receivers>
<max-bw>450</max-bw>
<src-id>666666 </src-id>
 </capture>
</endpoint>
```

```
</stream-map>
</mvv-conf-info>
```

4.5 MVV conference configuration

The configuration of a MVV conference is summarized as follows:

1) A conference focus receives a request to initiate a MVV conference for a group of user agents
2) The focus requests from each user agent its topology and capabilities information, using the MVV information event package, as described in 4.3.2.
3) Based on the MVV information of participants and the desired configuration for the conference, the focus generates a model of the conference, including one or more conference virtual spaces and a map of media streams. This process is discussed in section 5.7 .
4) To each participating user agent, the focus sends a SIP REFER request indicating the user agent to subscribe to MVV conference information for this conference. This process is detailed in 4.5.1.
5) The focus receives a SIP SUBSCRIBE request from each participant to subscribe to MVV conference information state, and answers with a MVV conference information document for this conference, following the procedure described in 4.4.
6) When all participants have received the MVV events, the conference can be initiated. The focus sends a SIP REFER request to each user agent, asking it to join the conference using SIP INVITE
7) The participating user agents send a SIP INVITE request to the focus and join the conference.

The process is illustrated in Figure 24 (see section 4.2).

The following sections discuss in detail some specific signalling scenarios, and certain steps of the signalling process.

4.5.1 Referring UAs to subscribe to MVV conference state

One necessary step in the configuration of MVV conferences involves the distribution of XML documents with the state of MVV conference information from the conference focus to all participating user agents. The challenge lies in the fact that the primary mechanism for distribution of XML documents to SIP UAs is the subscription/notification model, which has subscribing UAs actively requesting state information from a known notification server. However, at this point in the conference configuration process, user agents are not yet aware of the existence of the conference, much less of the conference URI required to access it – nor the fact that they are expected to subscribe to its state. Therefore, the conference focus needs a mechanism to push this state information to the user agents.

The answer lies in the SIP REFER method. A REFER request [RFC 3515]. “indicates that the recipient (identified by the Request-URI) should contact a third party using the contact information provided”. For most applications of REFER, contacting the third party involves sending a SIP INVITE request, but it is possible to have requests with other SIP methods – in our case, SIP SUBSCRIBE.

In order to set up a subscription to the MVV conference information of a given sorcery, the conference focus needs to send REFER requests whose Refer-To header value meets the following requirements:

- Point to the conference URI
- Specifies SIP SUBSCRIBE as the method for the referred request
- Requires the referred request to include an "Event" header with the value "mvv-conf-info".

Example:

Refer-To: sip:a@example.com;method=SUBSCRIBE?Event=conference

4.5.2 SDP Generation

At the end of the MVV conference configuration process, user agents join the conference by means of a SIP INVITE transaction. Media stream configuration is negotiated through the exchange of SDP bodies sent with each INVITE request and response. This negotiation process, as well as the generation of SDP messages, follows the Offer/Answer model [RFC3264]. However, the following rules also apply when generating an SDP message for a MVV conference:

1) A user agent MUST include in its SDP all media streams associated with that user agent in the stream map for the conference.

The stream map is an element within the mvv-conf-info document, which lists all media streams exchanged by user agents participating in a MVV conference, as described in section 4.4.1.6. SDP messages must be consistent with the information provided by the stream map, by including all media streams sent or received by the user agent joining the conference:

- The media streams sent by a user agent are represented as <capture> or <auxiliary-stream> elements within an <endpoint> element associated with that user agent in the stream map.
- The media streams received by a user agent are represented as <capture> or <auxiliary-stream> elements within the <endpoint> elements associated with other user agents in the stream map. Each <capture> or <auxiliary-stream-> has a <receivers> child element listing the user agents that will receive the corresponding media stream.

Note that media streams defined this way will usually be sent a single direction: streams sent by a user agent will have a "a=sendonly" attribute, and streams received by a user agent will have a "a=recvonly" attribute, in a SDP generated by that user agent.

2) The SDP configuration for a media stream MUST be consistent with the description of that stream in the conference stream map.

In a conference stream map, a <capture> or <auxiliary-stream> element representing a media stream can have child elements describing properties of the stream. These properties need to be preserved in the SDP description of that media stream:

- The media type in the SDP “m=” line needs to match the <media-type> child element in the <capture> or <auxiliary-stream> element.
- The SDP “label” attribute [RFC4574] needs to match the <label> child element in the <capture> or <auxiliary-stream> element.
- The SDP bandwidth (“b=”) attribute needs to match the <max-bw> child element in the <capture> or <auxiliary-stream> element.

3) <u>A user agent SHOULD NOT include in its SDP a media stream that is not present in the stream map for the conference, and associated with that user agent.</u>
4) <u>If the stream map provides a list of supported formats, SDP stream configuration MUST be consistent with that list.</u>

A stream map can include a <supported-formats> optional element, representing a list of encoding types supported by the terminals participating in a MVV conference. If this element is present, the encodings listed are the only ones that can be included in SDP messages for that conference. This can be useful for simplifying the negotiation process, particularly when intermediate media servers are involved.

4.5.3 MVV Conferences with media servers

4.5.3.1 MVV Conferences with non-mixing media servers

The role of non-mixing media servers in MVV conferences is explained in section 4.1.2.3. In short, they are intermediate nodes placed in the media path to route and replicate media streams. Non-mixing servers are usable whenever mixing is not an option for a given media stream (as is often the case with video in MVV conferences). The main advantage of these servers is their ability to replicate a media stream that needs to be received by two or more user agents, saving bandwidth at the access network of the user agent sending the stream. They also reduce the complexity in handling media streams at the endpoints, by allowing each user agent to send all streams to a single address (that of the media server) and delegate the task of routing streams to their final destination on the media server.

In this specification, it is assumed that all media streams in a MVV conference either go through a mixing media server or a non-mixing one. By default, a non-mixing server is assumed. The exception is the scenario of a MVV conference with just two participants, described in section 4.5.5, where no media server is required.

The non-mixing server is either part of the conference focus, or can communicate with the conference focus through means that fall outside the scope of this specification. Either way, the conference focus and non-mixing server meet the following requirements:

- The non-mixing server has access to the stream map of the conference, as defined in the mvv-conf-info document.
- The non-mixing server has a listening address/port pair to listen to every media stream in a conference. The conference focus has a table with these addresses and ports, and the identifiers of the streams assigned to them.

When the conference focus receives a SIP INVITE request from a user agent interested in joining a conference, it verifies that the SDP offer in that request includes all media streams assigned to that user in the conference stream map. It then generates a SDP answer with appropriate configurations for these streams, including listening addresses and ports corresponding to the media server. Media format information in the SDP answer is generated from the media parameters in the offer and any additional restrictions that have been specified in the stream map (such as the content of <supported-streams> elements). The conference focus sends the SDP answer in a SIP INVITE response.

If the conference is successfully initiated, the non-mixing media server will receive media streams from all user agents. For each media stream, the server consults its internal tables and forwards it to all user agents in the conference that are labelled as receivers of the stream.

4.5.3.2 MVV conferences with media mixing servers

In the previous section, we have described the use of non-mixing media servers in a MVV conference. It is also possible to have a mixing media server in such a conference. The requirements for the mixing server are similar to those defined for the non-mixing one – namely, it should be either co-located with the focus or communicating with the focus in a manner transparent to conference participants. Also, the mixer should have access to the stream map of the conference, and the conference focus should be aware of the mixer's listening addresses and ports, and the streams associated with them. To these requirements, the mixing media server adds the need to have access the information of the conference virtual space, whenever it is relevant to its mixing functions.

The process of mixing media boils down to generating new media streams that compose information from multiple media sources. As a consequence, a conference with a media mixer will have media streams whose content does not match any single stream offered by participating UAs. Because of this, any streams generated by a mixer need to be included in the mvv-conf-info. Notably, this includes generating positioning information to place the stream in the conference virtual space. Ideally, a media mixer should be able to generate mixed streams that are consistent with the virtual space perceived by any receiving participants.

4.5.4 Conferences with multiview displays

The use of multiview displays in a conferencing site to show different video views to each user at the site (as explained in 5.4) is a technique that can be used to improve eye contact in conferences with multiple users per site. When this kind of displays are used, they need to be reflected in any virtual space included in a mvv-conf-info document. The following rule applies:

A display that can render different views to each observing user is treated as **two separate <display> elements** in a MVV conference information document. Both <display> elements share the same media type and position, but have a different set of associated users.

This allows the conference configuration process to independently assign video views to each user observing the display.

4.5.5 Point-to-Point MVV Conferences

Although MVV conferences often have three or more participating user agents, it is possible to set up a point-to-point MVV conference with just two participants. An example scenario would be a session between two conferencing sites with multiple users each, where each conferencing site sends multiple captured views and renders video on multiple displays or a single multiview display. When this happens, the following rules can apply:

- The role of conference focus can be performed by a participating user agent. While this is technically true for any MVV conference, conferences with three or more participants are usually complex enough that the focus role is better delegated on an independent node. When only two participants are present, it will often be practical to have one of them act as the focus.
- Intermediate media servers, either mixing or non-mixing, are not needed for the conference. Such sessions typically have no need for media mixing or replication, so the presence of an additional media node is not justified.

4.5.6 Virtual View configuration

In a MVV conference, media streams are modelled as having a set capture position in the virtual space. When a media stream corresponds to a media capture from a physical device (i.e. is directly obtained from a camera or microphone), this capture position matches with that of the capture device. However, it is possible to have a stream composed from two or more media captures so that it appears to originate from an arbitrary position within a certain range. Such is the case with virtual views (ref to state of art chapter), a technique that can be used to improve eye contact between participants by placing a virtual camera at a specific location. The following steps should be taken to configure a virtual view in a MVV conference:

- The conferencing site offering a virtual view needs to describe it in its corresponding mvv-info document, by defining a <capture> element whose <position> has a 'position-type' of "variable". The <position-range> of this <capture> is used to describe the range of possible capture positions for the stream.
- The conference focus, when generating a virtual space for the conference, needs to select a fixed position for the virtual view within the offered range. It does so by defining a point in the virtual space for the <position> element of the <capture> corresponding to the virtual view, in the conference virtual space. Note that only the conference virtual space whose entity attribute corresponds with the SIP URI of the

site offering the virtual view (or the common virtual space, if that one is not defined) is taken into account by the offering site for the purposes of determining the origin of the virtual view.

- If the site offering a virtual view cannot or does not want to transmit the view using the origin defined by the conference focus, it should update its mvv-info state assigning a different <position-range> to the virtual view, or removing it altogether. The conference focus will then refresh the conference state and update the virtual space and stream mapping (including virtual view configuration) accordingly.

Note that the SDP entry of a media stream corresponding to a virtual view is indistinguishable from that of a regular video view.

4.5.7 Free Viewpoint Video configuration

So far, we have covered conferences where the capture position of media streams remains static, either because they correspond to physical capture devices with set locations, or because they are associated with virtual views whose perspective can be changed during conference configuration, but not mid-season. In terms of our signalling framework, these options correspond, respectively, to the “fixed” and “variable” position types defined for media streams in the MVV information document. With regards to MVV conference information documents, they are both treated as “fixed” position types, since they behave the same way once the conference starts. In this section, we discuss a different type of media stream characterized by the fact that its capture position can change in real time during an active conference. These media streams, which we call dynamic position streams, require a series of additional signalling parameters, described below.

The main application of streams with dynamic position is free-viewpoint-video, or a video stream which allows an observer to freely navigate around a scene by adjusting the position of the camera in real time. Although we are mostly interested in real-time communication use cases, the signalling mechanisms we propose can also be adapted to applications streaming pre-recorded FVV content, among other scenarios.

It should be noted that FVV streams are often composed from a large number of individual video streams. Although it is possible to transmit FVV by streaming each of these streams , or a subset of them (as demonstrated in [2011-Perez]), this is extremely costly in terms of network resources. Therefore, we will work off the assumption that the FVV stream is transmitted as a single media stream representing the rendered viewpoint that will be shown to the receiving user. In such a setup, the generation of this virtual view is performed at the sending user agent (i.e. the one capturing the FVV stream), rather than at the receiver. This introduces the need for some control channel to allow the receiving side to ask the sending UA to change the rendered viewpoint, and makes the application even more sensitive to network delay than regular video communications. On the other hand, the bandwidth requirements of FVV video transmitted this way do not diverge substantially from those of a conventional video stream.

The signalling of a MVV stream in a conference can be summarized in the following steps:

1) Describe the MVV stream in the mvv-info document of the sender
2) Describe the MVV stream and associated auxiliary streams in the mvv-conf-info of the conference
3) Describe the MVV stream and associated auxiliary streams in the SDPs of the conference.

4.5.7.1 FVV streams in mvv-info documents

A user agent announcing one or more free-viewpoint-video streams in its mvv-info document needs to identify the streams as dynamic position streams, delimit the valid ranges for camera position and capture position, and provide identifiers for any auxiliary streams. In addition, it needs to indicate support for any specific media format used by the auxiliary streams. This is summarized as follows:

- In the <capture> element associated with the FVV stream:
    - 'position-type' attribute must have a value of "dynamic".
    - The <position> attribute must include one or more <position-range> elements.
    - If auxiliary streams for conveying position information are used, the <capture> element must include a <position-stream-id> element.
    - If auxiliary streams for controlling capture position of the FVV stream are used, the <capture> element must include a <control-stream-id> element.
    - If the area of capture can vary, the <capture> element must include one or more <capture-range> elements.
- In the <supported-elements> in the <mvv-capabilities> element of the UA sending the FVV stream, there must be one or more supported <encoding> elements for a media format suitable for position information or control auxiliary streams associated with FVV. This requirement must also be met at any UA interested in receiving the FVV stream.

Note that no standard media formats for the required auxiliary streams exist currently. The definition of such formats falls outside the scope of this document.

An example of the relevant parts of a mvv-info document including a FVV stream is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<mvv-info (…)>
<capture-list>
<capture id="c-fvv">
  <media-type>video</media-type>
  <position position-type="dynamic">
    <position-range>
      <point x="-2000" y="-2000" z="300"/>
      <point x="-2000" y="2000" z="300"/>
      <point x="2000" y="-2000" z="300"/>
      <point x="2000" y="2000" z="300"/>
      <point x="-2000" y="-2000" z="1800"/>
      <point x="-2000" y="2000" z="1800"/>
      <point x="2000" y="-2000" z="1800"/>
      <point x="2000" y="2000" z="1800"/>
    <position-range>
  <position-stream-id>c-fvv-pos</position-stream-id>
  <control-stream-id>c-fvv-con</control-stream-id>
  </position>
  <capture-area>
    <capture-range>
      <point x="-1000" y="-1000" z="600"/>
      <point x="-1000" y="1000" z="600"/>
      <point x="1000" y="-1000" z="600"/>
      <point x="1000" y="1000" z="600"/>
      <point x="-1000" y="-1000" z="1500"/>
      <point x="-1000" y="1000" z="1500"/>
      <point x="1000" y="-1000" z="1500"/>
      <point x="1000" y="1000" z="1500"/>
    <capture-range>
  </capture-area>
</capture>
</capture-list>

<mvv-capabilities>
  <supported-formats>
   <encoding media=application name=fvv-position/>
   <encoding media=application name=fvv-control/>
 <supported-formats/>
</mvv-capabilities>
```

4.5.7.2 FVV streams in mvv-conf-info documents

A conference focus configuring a conference with one or more free-viewpoint-video streams needs to include

The inclusion of one or more free-viewpoint-video streams in a conference affects both the virtual space and the stream map of the mvv-conf-info document for the conference. The virtual space has to identify the stream as a dynamic position stream, and define the valid ranges for camera position and capture position. Meanwhile, in the stream map, the FVV stream has to be associated with its auxiliary streams, and the user agent in charge of controlling the position of the FVV needs to be identified. Also, the auxiliary streams need to be defined independently in the stream map. Finally, the media formats for the auxiliary streams need to be described as required for the conference in the stream map. This is summarized as follows:

- In the <capture> element associated with the FVV stream in the virtual space:
    - ‘position-type’ attribute must have a value of “dynamic”.
    - The <position> attribute must include one or more <position-range> elements.
    - If the area of capture can vary, the <capture> element must include one or more <capture-range> elements.
- In the <capture> element associated with the FVV stream in the stream map:
    - If a UA is responsible for the control of the FVV stream, it should be included in a <receiver> element with a <role> child element of value “controller”.
    - If auxiliary streams for conveying position information are used, the <capture> element must include a <position-stream-label> element.
    - If auxiliary streams for controlling capture position of the FVV stream are used, the <capture> element must include a <control-stream-label> element.
- Any auxiliary stream must be represented in the stream map by its own <capture> element in the stream map, including:
    - An <auxiliary-function> element with value of either “control-stream” or “position-stream”.
    - An <associated-stream-label> containing the label of the associated FVV stream.
- In the <supported-elements> in the <stream-map> element of the UA sending the FVV stream, there must be one or more supported <encoding> elements for a media format suitable position information or control auxiliary streams associated with FVV.

An example of the relevant parts of a mvv-conf-info document including a FVV stream is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>

<mvv-conf-info (…)

<virtual-space entity="sip:conf1@example.com">


<capture-list>
  <capture id="c-fvv" entity="sip:m@example.com">
   <media-type>video</media-type>
   <label>c-M-B</label>
   <position position-type="dynamic">
     <point x="-1000" y="1730" z="1200"/>
    <position-range>
      <point x="-2000" y="-2000" z="300"/>
      <point x="-2000" y="2000" z="300"/>
      <point x="2000" y="-2000" z="300"/>
      <point x="2000" y="2000" z="300"/>
      <point x="-2000" y="-2000" z="1800"/>
      <point x="-2000" y="2000" z="1800"/>
      <point x="2000" y="-2000" z="1800"/>
      <point x="2000" y="2000" z="1800"/>
    <position-range>
   </position>
   <capture-area>
    <point x="-350" y="-200" z="850"/>
    <point x="350" y="200" z="850"/>
    <point x="-350" y="-200" z="1300"/>
    <point x="350" y="200" z="1300"/>
    <capture-range>
      <point x="-1000" y="-1000" z="600"/>
      <point x="-1000" y="1000" z="600"/>
      <point x="1000" y="-1000" z="600"/>
      <point x="1000" y="1000" z="600"/>
      <point x="-1000" y="-1000" z="1500"/>
      <point x="-1000" y="1000" z="1500"/>
      <point x="1000" y="-1000" z="1500"/>
      <point x="1000" y="1000" z="1500"/>
    <capture-range>

   </capture-area>
  </capture>
</capture-list>

<stream-map>
  <endpoint entity="sip:m@example.com">
    <supported-formats>
      <encoding media=application name=fvv-position/>
      <encoding media=application name=fvv-position-control/>
    <supported-formats/>

    <capture id="c-fvv">
      <media-type>video</media-type>
      <label>c-FVV</label>
      <receivers>
        <receiver entity="sip:b@example.com">
          <role>controller</role>
        </receiver>
      </receivers>
      <position-stream-label>c-FVV-POS<position-stream-label>
      <control-stream-label>c-FVV-CON<control-stream-label>
    </capture>

    <auxiliary-stream id="c-fvv-pos">
      <media-type>application</media-type>
      <auxiliary-function>position-stream</auxiliary-function>
      <label>c-FVV-POS</label>
      <associated-stream-label>c-FVV</associated-stream-label>
      <receivers>
        <receiver entity="sip:b@example.com">
          </receiver>
      </receivers>
      </auxiliary-stream>
  </endpoint>

  <endpoint entity="sip:m@example.com">

    <auxiliary-stream id="c-fvv-con">
      <media-type>application</media-type>
      <auxiliary-function>control-stream</auxiliary-function>
      <label>c-FVV-CON</label>
      <associated-stream-label>c-FVV</associated-stream-label>
      <receivers>
        <receiver entity="sip:m@example.com">
        </receiver>
      </receivers>
    </auxiliary-stream>
  </endpoint>
</stream-map>
</mvv-conf-info>
```

4.5.7.3 FVV in SDP

The addition of a FVV stream requires no special extensions at SDP level. The FVV stream, as well as any auxiliary stream defined for it, must be present in the SDP of the conference participants sending or receiving FVV media, and its configuration must match the information of the mvv-conf-info document, as with any other stream.

Note that, depending on the media format used for the auxiliary FVV streams, the definition of a new SDP field might be required in order to support these streams. This should be considered part of the process of defining that media format and, as such, outside of the scope of the current document.

4.5.8 Quality of Service

A MVV conference can benefit from a network infrastructure capable of providing quality of service. Like any video communication session, these conferences are highly sensitive to delay or packet losses introduced by the network, particularly when features like free-viewpoint video or virtual views are used (since these tend to add their own delay component). In this section we discuss the signalling implication of configuring a conference with QoS.

In SIP-based sessions, QoS reservations are performed at the SDP level, with intermediate network nodes examining SDP fields in order to determine which media streams require dedicated network resources. Typically, bandwidth reservations are performed by including a “b=” SDP field at either the session or media level. The signalling framework defined for MVV conferences preserves this mechanism, and does not affect the content of SDPs in any way that could interfere with QoS infrastructure. That said, the conference focus needs to be aware of any QoS requirements for the session in order to indicate user agents what resources to reserve.

Information about QoS requirements for individual media streams and the conference as a whole can be exchanged between user agents and the conference focus by means of the mvv-info and mvv-conf-info documents. User agents can inform of the bandwidth required by their captured streams by including a <max-bw> element within the <capture> associated with each stream; the content of this element determines the value that a “b=AS:” field in SDP would need to have. Likewise, the conference focus communicates this information by including <max-bw> elements in the respective <captures> within the stream map.

5 Analysis and generation of conference virtual spaces

In the previous chapter, we have defined a **virtual space** as a common coordinate system where elements in participating conferencing sites are distributed for the purposes of one conference, including representations of the users, display devices, and capture devices at each site. By analysing the virtual space of a session, one can determine whether the session can support certain features, like eye contact or real-size video. In this session, we describe how to perform this analysis, and the conditions that a virtual space has to meet in order to provide eye contact and spatial faithfulness in a conference.

To conclude, we define a series of guidelines for generating a virtual space, which can be used by a conference focus when configuring a multiview video conference.

5.1 Definition of session properties

Certain properties of a multimedia session are affected by the geometry of the virtual space associated to that session. These properties include gaze awareness, spatial faithfulness, video scale, and video framing. We describe them in the following sections.

5.1.1 Gaze awareness and spatial faithfulness

One of the main advantages of video-based communication systems is their ability to convey nonverbal communication cues, such as gaze, facial expressions and hand gestures, between users. These cues improve communication by transmitting information on a variety of topics, including attitude, degree of attention, or references to nearby objects. However, non-verbal communication often relies on the spatial disposition of speakers and the people and objects surrounding them, and these spatial relationships can be distorted or lost over a remote communication session. In this section, we discuss how to characterize the ability of a communication system to preserve nonverbal cues in general, and gaze direction in particular.

[2002-Monk] introduced the concept of gaze awareness in a conferencing system, defining it as the ability to identify the direction of a remote user’s gaze. Three levels of gaze awareness are possible, depending on whether subjects are aware of being looked by others, whether they can determine the general direction of their gaze, and whether they can identify the exact object being looked at. [2005-Nguyen] generalized gaze awareness it to include other nonverbal cues reliant on spatial relationships, introducing the notion of spatial faithfulness. A system is considered spatially faithful when it preserves the spatial relationships between the people taking part in a communication session and any objects around them. Three levels of spatial faithfulness exist, equivalent to those defined for gaze awareness. Finally, [2012-Moubayed] used a similar classification, defining the concept of gaze faithfulness and three levels of increasing faithfulness.

We have refined and adapted these concepts to work in the context of our conferencing model, resulting in the following definitions:

- **Gaze Awareness**: Gaze awareness describes the level to which a user in a conference perceives the direction of gaze of a given remote user. There are three possible levels of gaze awareness, in increasing order: mutual gaze awareness, gaze orientation awareness, and gaze object awareness.

- **Mutual Gaze Awareness:** Also known as **Eye Contact**. A user has mutual gaze awareness with relation to a remote user in a video session if:
    - The user can perceive when the remote user is looking at him.
    - The user can perceive when the remote user is not looking at him.
- **Gaze Object Awareness:** A user has gaze object awareness with relation to a remote user in a video session if:
    - The user can identify which object or person the remote user is looking at.
- **Spatial Faithfulness**: A video session is spatially faithful if:
    - All users have gaze object awareness to one another.

Note that under this definition, gaze awareness has the following properties:

- Relative to a session. Gaze awareness is not an intrinsic quality of conferencing terminals, but a property of a multimedia session, which depends on the characteristics of participating terminals, as well as session configuration.
- Defined between a pair of users. At a multimedia session, it is possible for some users to have gaze awareness of different levels between them while others lack it. A given user may have gaze awareness to certain remote users, but not to others.
- Single direction. At a multimedia session, gaze awareness from one user to another does not imply the same in the opposite direction; the remote user may have an equal level of gaze awareness towards the local one, but also a higher or lower one.

Likewise, the definition implies the following properties for spatial faithfulness:

- Relative to a session. Like gaze awareness, spatial faithfulness is a property of multimedia sessions, affected by terminal properties and session configuration.
- Affects all users in session. Every user in a session must be able to tell the object of attention of all remote users for the session to be considered spatially faithful.

Both gaze object awareness and spatial faithfulness are binary properties: either a user or session has them, or they don't have them. However, for mutual gaze awareness and gaze orientation awareness, we can define a **Gaze Orientation Error** as a variable that quantifies the deviation in gaze direction of the remote user, in sessions with imperfect gaze awareness. To put it more formally:

- **Gaze Orientation Error:** Also known as **Gaze Error**. In a video session, the gaze orientation error from one local user to a remote user is the angle, in degrees, in which the direction of gaze observed for the remote user deviates from the local user, when the remote user is looking straight at the image of the local user.

It is possible to have mutual gaze awareness from one user to another in the presence of a non-zero gaze error. This is due to the fact that gaze direction is not perceived as a line, but in the form of a cone, so there is a range of gaze error angles for which a local user will still correctly perceive when the remote user is gazing at him.

Previous work in measuring the form and size of the cone and gaze is discussed in the following subsection.

5.1.1.1 The cone of gaze

The human perception of gaze direction is a topic of particular interest for the design of video conferencing systems, as it determines the requirements that these systems need to meet in order to support gaze awareness. Studies of this subject show that, for the purposes of eye contact, gaze direction does not behave as a line, but as a region of space that can be approximated as a cone.

[2007-Gamer] first characterized gaze direction as a “cone of gaze”, after performing experiments measuring gaze detection for different viewing distances, and observing that the angular range in which a viewer feels looked at remained relatively stable. Previous studies, like [2002-Chen], had focused on measurements of gaze detection from fixed distances.

An important discovery regarding the behaviour of gaze detection was made by [2002-Chen], who noticed that observer sensitivity to eye contact was drastically reduced when a remote user looked below the camera. That is, the maximum gaze orientation error for which observers perceived eye contact was much higher (with a 50% gaze detection rate for errors of 10º) if the remote user looked downwards than if he was looking in any other direction. Based on these results, the authors recommended desktop conferencing configurations placing cameras above the monitor, with a gaze error of up to 5º.

However, later studies have failed to identify this asymmetry of eye contact. [2009-VanderPol] obtained eye contact sensitivity values that remained stable regardless of the direction of gaze error. Meanwhile, [2010-Van Eijk] observed a higher tolerance to gaze errors on the vertical axis than on the horizontal one, but with a much less pronounced difference than in [2002-Chen], and only slightly favouring the down direction relative to the up direction.

Table 23 summarizes observed values for cone of gaze dimensions, with the horizontal and vertical width representing, respectively, the angular width of the observed eye contact area at which the perceived eye contact rate is of 50%. The vertical offset indicates the average error for observer estimation of the vertical component of gaze direction, since gaze direction as perceived by users was systematically below actual gaze direction. No significant vertical offset was identified. Note that vertical width and offset values are missing from [2007-Gamer] because no measurements on the vertical axis were performed in that study.

Table 23 Eye contact detection measurements

Source	Viewing Distance	Eye contact angle (50% contact)		
		Horizontal width	Vertical Width	Vertical offset
2007-Gamer	1m	8.3º	-	-
2009-Van der Pol	1.25m	6.8º	6.8	-0.4º
2010-van Eijk	2m	4.7º	7º	-0.5º
2002 Chen	2.4m	2º	12º	-5º

These results show moderate variance from one experiment to another, which is partially justified by changes in viewing distance – the “cone” of gaze tends to become narrower with distance. That said, it is hard to reconcile the values from [2002-Chen] with those of the other authors, suggesting either one or the others are somehow flawed. Given that the values of the remaining experiments are relatively consistent, we are inclined to favor these results over Chen's.

Based on these experiments, we recommend using the guidelines from Table 24 to evaluate the quality of eye contact from one user to another, for a given gaze direction error. Keep in mind that the results from Table 23 refer to width of eye contact region (which is roughly twice the maximum gaze error for the region), and that these values describe the region in which users perceive eye contact 50% of the time, which is insufficient for a quality conferencing experience.

Table 24 Guidelines for eye contact evaluation

Eye contact quality	**Gaze error**
No eye contact	>3º
Poor eye contact	Between 1.5º and 3º
Acceptable eye contact	<1.5º

We consider there is no eye contact if the eye contact detection rate is lower than 50%. At the "poor eye contact" range, the detection rate is at least 50%. Finally, an acceptable level of eye contact requires detection rates of 75% or higher.

Please note that these guidelines are based off approximated figures, due to the variance of available experimental data. Further experiments would be needed in order to more accurately characterize the relationship between eye contact quality and errors in gaze direction. Still, we believe that Table 24 can be a useful tool to quickly evaluate gaze awareness in a session.

As a refinement to these guidelines, and based on known measurements of vertical offset of the gaze cone, we would suggest to apply a 0.5º adjustment to the vertical component of gaze errors. That is, a gaze error of 2º in the 'up' direction would be treated as a 2.5º error, whereas the same error in the 'down' direction would count as an error of just 1.5º. This is consistent with the fact that all experiments show more error tolerance when remote users are looking downwards.

5.1.2 Video scaling and framing

One of the most effective ways to improve sense of presence in a session is conceptually very simple, but not necessarily easy (or cheap) to implement: to use larger videos. In this section, we discuss image size and its impact in telepresence, with an overview of possible configurations and the trade-offs they involve.

Generally, user experience in a multimedia session improves significantly by increasing the size at which remote sites are displayed, up to the point where it is possible to observe all relevant remote users and objects at real size. However, this also brings a drastic increase in system cost due to the need for huge screens and the bandwidth consumption of high resolution video streams. Thus, life-sized video is mostly relegated to high end conferencing rooms at the moment. Also, and of particular note for our work, it is typically very hard to combine life-sized video with novel display techniques, like multiview displays.

When a complete, life-sized representation of remote sites is not possible (as is often the case, typically due to limitations on display size), the system has to reach a compromise regarding two properties : **video scale**, and **video framing**.

Video scale refers to the size of displayed objects relative to the originals. Displaying a video at a scale below its real size often results in a loss of image definition and, more importantly, it breaks the geometry of the session space. Objects at a downscaled remote site appear smaller and closer to each other, and the spatial relationships between them and other objects in the session cannot be accurately estimated.

As a consequence, the use of video scaling has a negative effect in the level of gaze awareness in a session (see 5.1.1). Specifically, a user cannot have gaze object awareness with a remote user if the video for the remote user is scaled. This also means that a session where video scaling is used cannot be spatially faithful.

It is rare to scale videos to be larger than life in conferencing environments, since there is little practical reason to do so, outside of having a remote user talking to a class or auditorium. In those scenarios, the priority is to make the user viewable by observers that are far from the screen, and eye contact is not a concern.

Video framing [2009-Nguyen] refers to the portion of an object that is displayed to the observer. In conferencing, the level of framing determines whether the whole bodies of remote users are displayed, or only parts of them, such as their faces. We define the following framing types for a remote conferencing user:

- Full Body Framing – The whole user's body is shown.
- Upper Body Framing – The upper body of the user is shown. Typically used for users sitting behind a table.
- Head Framing – Only the head and shoulders of the user is shown.

Head framing provides the minimum acceptable configuration for a video session, since at the very least one should be able to observe the full face of a remote user for the use of video to be worthwhile. Upper body framing expands the previous configuration by showing the user's torso, arms and, crucially, his hands. Hand gestures are a rich form of nonverbal communication, providing a large variety of cues, which makes videos with this kind of framing much more expressive than those showing just the heads. As for full body framing, it is not particularly useful in typical conferencing scenarios where the users remain sitting behind a table, since most of the user's body would be covered anyway, so it should be applied in cases where the user is standing or moving around. The addition of the rest of the body does not bring as much of an improvement in communication as the jump from head framing to upper body framing, though, so this framing should be reserved for specific applications, like watching the remote user dance, or perform some exercise.

The exact dimensions required to display each type of framing at a 1:1 scale vary with the size of the user and the range of movements expected for the session. In general, a frame size of 25x30 cm is a good baseline for head framing, and one of 50x70 cm is enough for upper body framing. As for full body framing, its requirements are the most dependent on user height and type of physical activity, but a 90x180 cm frame should be enough for average users standing still, whereas moving users could require about 180x180 cm frame.

Both scaling and framing can be used to translate a large scene into a remote display, and the techniques can be applied separately, or combined with one another to different degrees. To our knowledge, there has been no study comparing the effects of scale and framing on user experience and feeling of presence in video conferences. Given this lack of empirical evidence, we will work with the following assumptions:

1) Full body framing is impractical for most video conferences
2) The benefits of using upper body framing over head framing outweigh the advantages of life-sized video over downscaled video
3) It is undesirable to scale video down to extremely small sizes

The first assumption is justified by the predominance of conferences featuring users sitting around tables. In the second point, we consider that the added expressivity gained by showing users' hands and body is a more important feature for a communication session than the increased realism of users displayed in real size. This becomes a harder call if the use of head framing and life-sized video allows the session to feature spatial faithfulness (which is not compatible with scaled video), but ultimately, the usefulness of spatial faithfulness is limited when hand gestures cannot be displayed. Finally, the third assumption acknowledges that for some terminals (mobile ones in particular), the reduced screen size does not really support showing more than the remote user's face without reducing the video to an unacceptable scale.

Based on these assumptions, we have defined these guidelines to process video streams when screen size does not allow for life-sized rendering:

- Full body framing should be reserved for video applications that specifically call for it, e.g. a session for a remote dance lesson.
- In conferencing sites and desktop terminals, video should be presented with upper body framing at the largest scale possible

- In mobile terminals, video should be presented with head body framing unless screen size and resolution are exceptionally high.

5.2 Eye contact between two users

Eye contact between two users in a session can be identified by examining the disposition of elements in the virtual space describing that session.

Consider a session with the following elements:

- Two participating sites, A and B
- Site A has the following elements:
  - User **uA**
  - Display **dA**
  - Captured stream **cAB**, with the view of user uA that will be displayed to user uB
- Site B has the following elements:
  - User **uB**
  - Display **dB**
  - Captured stream **cBA**, with the view of user uB that will be displayed to user uA

There is eye contact between users uA and uB for this session if, in the common session virtual space defined for the session, the following conditions are met:

a) The line defined by the positions of users uA, uB contains the capture positions cAB, cBA

b) The rendering of user uA in display dB, **uA_dB**, is placed so that its position in the virtual space is contained in the line defined by (uA,uB).

c) The rendering of user uB in display dA, **uB_dA**, is placed so that its position in the virtual space is contained in the line defined by (uA,uB).

That is, there is eye contact if users, cameras, and displayed user images are aligned in the virtual space.

Conditions b) and c) are affected by how views are rendered on each display. For example, a view can occupy the whole surface of a display, the right half, or just a window. In a session virtual space, this information is included in the association of views to a display. In some cases (e.g. when generating a virtual space), it may be desirable to know whether eye contact is possible between users before rendering details have been defined. The following condition determines whether conditions b) and c) can be satisfied for a given virtual space:

d) The line defined by the positions of users uA, uB crosses the area of displays dA, dB

That is, it is possible to display user images so that they are aligned with users and cameras in the virtual space.

In practice, when the line (uA,uB) crosses near the edges of a display area, it won't be possible to display the image of a user's face without cropping it. Thus, for rule d), it is better to use display area minus a safety margin of at least 10 centimetres on each side (or less, if the views are downscaled).

The following sections provide examples of sessions with and without eye contact, and their corresponding virtual spaces. Note that, for the sake of clarity, the diagrams in these examples only show virtual spaces in 2 dimensions. In real deployments, virtual space analysis needs to take into account all 3 dimensions of space. Notably, this means that the height of users participating in a session affects eye contact properties of a session; thus, for an optimal experience, user agents need to be able to dynamically measure user height, or seats at a site need to be adjusted so that user heads remain at a constant level.

It is important to emphasize that eye contact is not necessarily symmetric, that is, it is possible to have sessions where user uA correctly perceives the direction of user uB's gaze, but uA's gaze is displayed incorrectly to uB. The example in section 5.2.2 illustrates this.

5.2.1 Example 1: Eye contact between 2 users, good alignment

The first example shows a session with proper eye contact between two users. In Figure 32, we can see the virtual space of such a session. Elements from site A (in red) include user uA, capture cAB, and display dA; elements from site B (in blue) include user uB, capture cBA, and display dB.

The figure shows that this session virtual space meets conditions a) and d) (as defined in the previous section). That is, the cameras and screens are aligned with users in the virtual space. The **grey dotted line** includes both users and cameras, and crosses the displays.

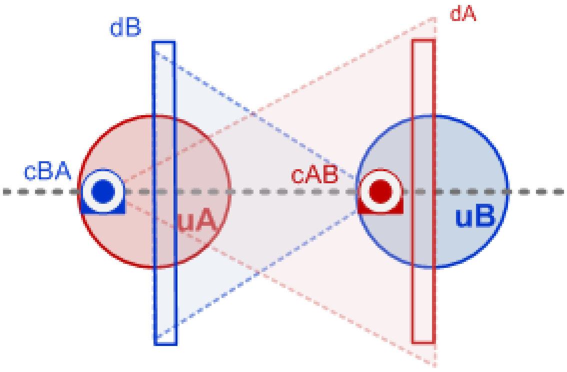

Figure 32 Virtual Space of session with eye contact

Due to the overlap of elements in the virtual space, it is not possible to clearly show the position of displayed user images in the previous figure. Figure 33 shows separately the elements from each conferencing site, including the positions of user renderings on screen, uB_dA and uA_dB.

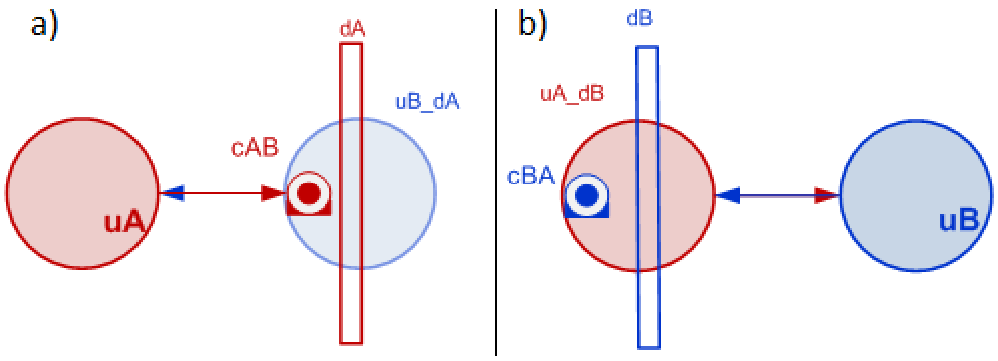

Figure 33 Eye contact at a) site A, b) site B

The figure also represents the direction of user gazes with colored arrows (red for user uA, blue for user uB). At each site, these arrows describe the direction of the gaze of the local user and of the rendered remote user, *when both users in the session are looking straight at each other.* The gaze arrows will match if and only if the image of the remote user is aligned with the camera capturing the local user. In this case, all elements are properly aligned, and the eye contact between users is perfect.

5.2.2 Example 2: Eye contact between 2 users, misaligned camera

The second example illustrates a session which lacks eye contact due to a camera misaligned with the users. As before, we have a session with two sites, A and B, users uA and uB, cameras cAB and cBA, and displays dA and dB.

Figure 34 shows the virtual space of this session, where all elements are aligned except for the camera capturing view cBA (that is, the view captured at site B, which shows user uB and is rendered to user uA). In this case, this camera is placed at the edge of display dB, which is a common occurrence in practical scenarios, since cameras cannot typically be placed directly in front or behind a screen, barring use of virtual views.

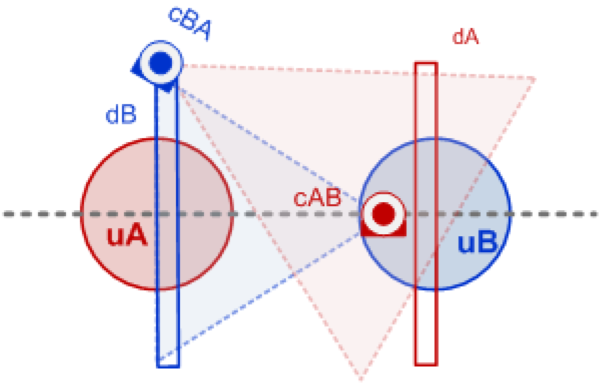

Figure 34 Virtual space of session with misaligned camera

Figure 35 shows separately the elements from each conferencing site for this session, including the positions of user renderings on screen, uB_dA and uA_dB, and the direction of user gazes (red arrow for user uA, blue arrow for user uB). As in the previous example, the arrows represent gaze direction of the local user and rendered remote user, when users look directly at each other.

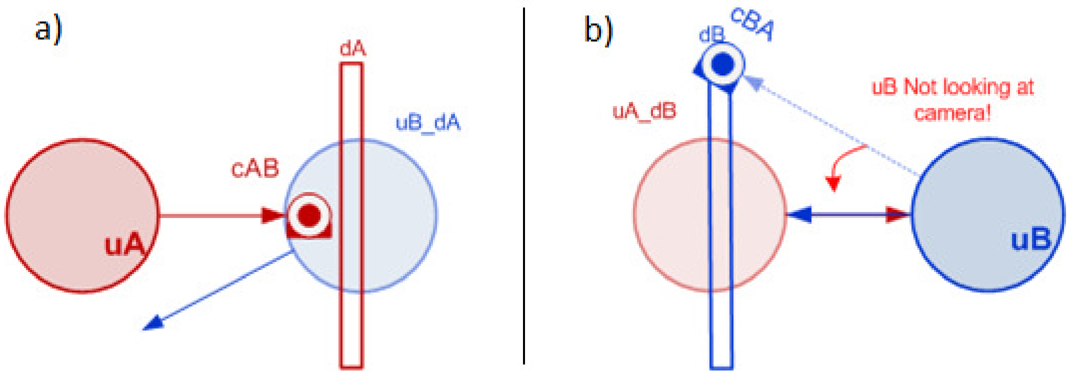

Figure 35 a) Gaze error at site A, b) eye contact at site B

In this case, there is a gaze error produced by the incorrect position of camera cBA. At site B, when user uB looks straight at the rendering of user uA on screen, uB's gaze is not directed at camera cBA. Because of this, at site A, user uA sees the rendering of uB as not looking back at uA. That is, this session setup results in a lack of eye contact for site A.

Note that there is still correct eye contact at site B, since the camera and display at site A are properly aligned. The misalignment of a camera at one site only results in eye contact errors at the remote site.

5.2.3 Determining gaze error between two users

When users, cameras, and rendered views are not correctly aligned in a session, this results in a gaze error, or lack of eye contact between two users. This error can be measured as the angular distance between the ideal gaze direction and the direction of the gaze of a rendered user. We can determine the gaze error between any pair of users in a MVV session by analysing the virtual space defined for the session, as we describe below.

The gaze error **error(uA,uB)** that a user **uA** observes in the rendering of a remote user **uB** for a given session is the angle formed in that session's virtual space by:

- The line (uA, uB) formed by the position of the two users
- The line (uB, cBA), formed by:
  - user uB
  - the position of the camera **cBA** capturing a view of uB for uA

This is illustrated in Figure 36, which shows the virtual space from the example in 5.2.2 (of a session lacking eye contact due to a misaligned camera), with the gaze error, error(uA,uB)=**α**,in green.

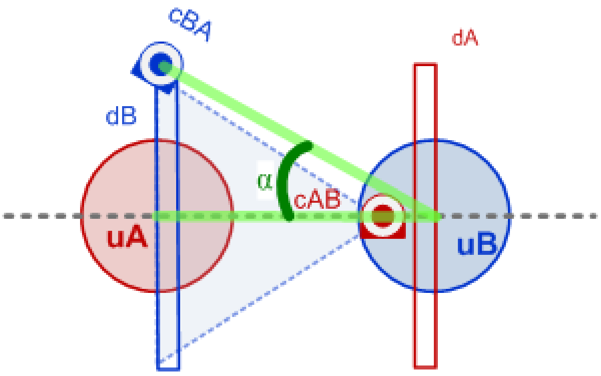

Figure 36 Determining gaze error from uB to uA

It is important to note that gaze errors can be asymmetric: for any pair of users uA and uB, the error observed by uA , error(uA,uB) can be different from the error observed by uB, error(uB,uA). In this example session, user uB would observe correct eye contact, so error(uB,uA) would be zero.

5.2.4 Determining minimum eye contact error of a conferencing site for a given view-user pair

Although the actual gaze error incurred when rendering a view depends on the specific configuration of each session (i.e. its session virtual space), its value is constrained by the intrinsic properties of each participating site. By analysing the spatial relationships between the components of a conferencing site outside of any session context (i.e. its local virtual space), we can determine the minimum gaze error that can be introduced when transmitting a view of a local user to a remote site.

Consider a conferencing site with a user **uB**, a display **dB** and a camera **cBA** capturing a view of uB. The gaze error caused when transmitting view cBA to a remote user will depend on how user, display and camera are aligned, as well as where this remote user is rendered in the local display. The first parameters are intrinsic to the conferencing site, whereas the rendering position of remote users varies with session configuration.

Figure 37 illustrates this for a conferencing site where user, display and camera are aligned so as to enable zero gaze error in the best case scenario. This optimal configuration will be attained in sessions where a remote user is rendered in a position that is aligned with user uB and camera cBA, as shown in example a) of the figure. In that scenario, the image of the remote user **uA_dB** is centred in the point where the line (uB, cBA) intersects with the display area dB. We call that point the optimum rendering point.

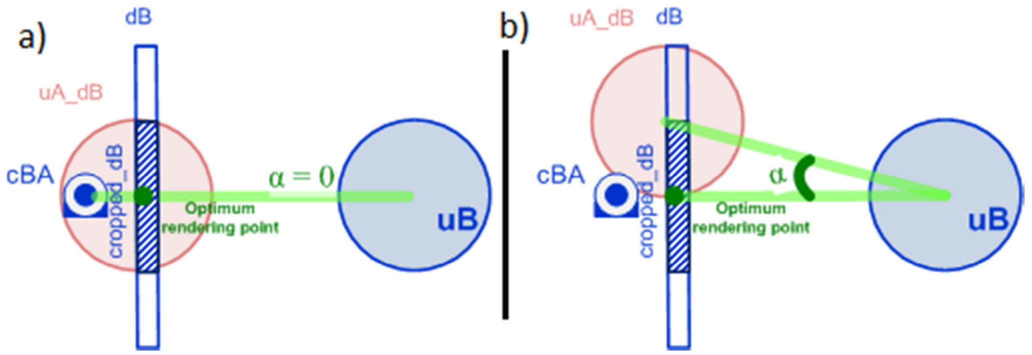

Figure 37 Site with zero minimum gaze error, optimal configuration (a) and non-optimal configuration (b)

A theoretical minimum gaze error of zero does not guarantee perfect eye contact in an actual session, though. This is shown in example b) of Figure 37, in which the same conferencing site of the previous example is rendering the remote user at a location other than the optimum rendering point. In this case, user uB is not looking straight at camera cBA when talking to the remote user, incurring in a gaze error *error(uA,uB)*=**α**

For a scenario where the conferencing site has a nonzero minimum gaze error, the user, camera and display are not perfectly aligned. This is illustrated in Figure 38. In these cases, the line between user and camera (uB,cBA) either does not cross the display area, or it does but the intersection is so close to the edges of the area that it is not possible to render the face of a remote user in that position. Thus, any valid rendering position for the remote user will fall outside of the user-camera line. In this scenario, we define the <u>optimum rendering point</u> as the point of the cropped display area (i.e. display area minus safety margin) where a remote user can be rendered so that the gaze error is minimized. To find this point:

1) Determine the point, **P**, where the user-camera line (uB, cBA) intersects with area of display, or with the plane defined by the area of display.
2) The optimum rendering point is the point of the cropped display area closest to point P.

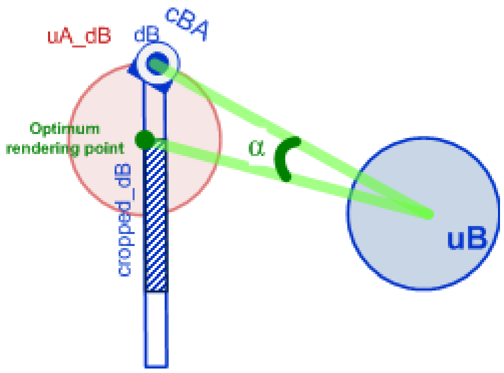

Figure 38 Minimum gaze error at site B for view cBA

The **minimum gaze error** for this conferencing site, user uB and view cBA is defined as the gaze error for the best case scenario of a video conference where view cBA is transmitted from this site to a remote user. Its value is the angle formed by:

- The line (uB, cBA) formed by user and camera.
- The line formed by user uB and the optimum rendering point assigned to the remote user, user uB, and view cBA.

5.3 Eye contact with multiple users per site (single view)

When multiple users are present at a conferencing site, it is no longer possible for all local users to simultaneously maintain eye contact with a remote user, unless each local user is shown a different view. This is illustrated in the following example: consider a session between two sites, A and B, where site A has two users **uA1** and **uA2**, and site B has a single user **uB**. Each site has a single camera (**cAB** at site A, **cBA** at site B) and display (**dA** at site A, **dB** at site B). The virtual space for this session is shown in Figure 39.

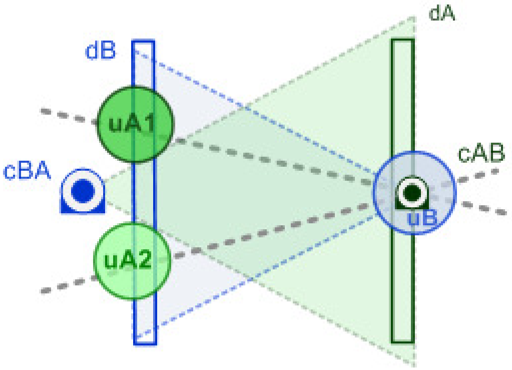

Figure 39 Virtual space of session with multiple users at a site (no eye contact)

Note that for this session, the camera at site A is aligned in the virtual space with uA1, uB and dA, so that user uB will perceive correct eye contact from user uA1. Likewise, this camera is also aligned with uA2, uB and dA, so user uB will also correctly perceive user uA1's gaze. However the same does not apply in the opposite direction: the camera at site B is placed halfway between users uA1 and uA2, and cannot be aligned with neither user. As a consequence, both users uA1 and uA2 will perceive an error in gaze direction from uB. Figure 40 shows how gaze direction is observed at each site, when user uB is looking at uA1, and both uA1 and uA2 are looking at uB. Though not shown in the figure, the session will present an equal gaze error in the opposite direction when user uB is looking at uA2.

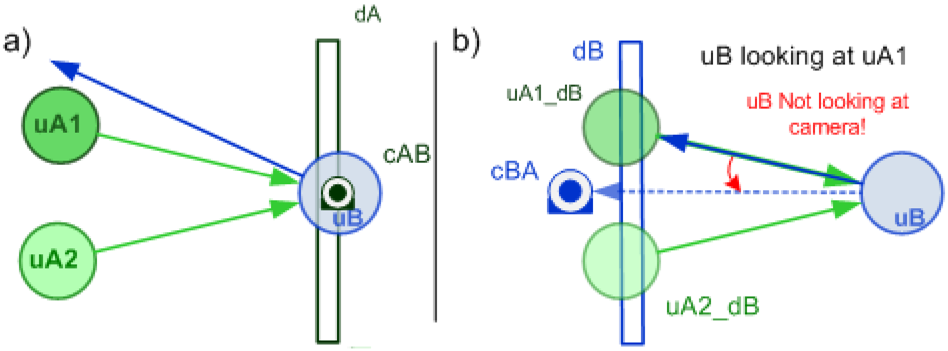

Figure 40 a) Gaze direction at site A; b) Gaze direction at site B

This gaze error is also shown in Figure *41*, which shows the display dA as observed by users uA1 (top left) and uA2 (top right). Again, both uA1 and uA2 are looking at uB, whereas uB is looking at uA1. The bottom row shows the same viewpoints in a session that properly conveys eye contact (described in 5.4), for comparison. Note how both uA1 and uA2 see the same perspective of uB, and how uA1 sees uB gazing at his left, even though uB is actually looking straight at uA1.

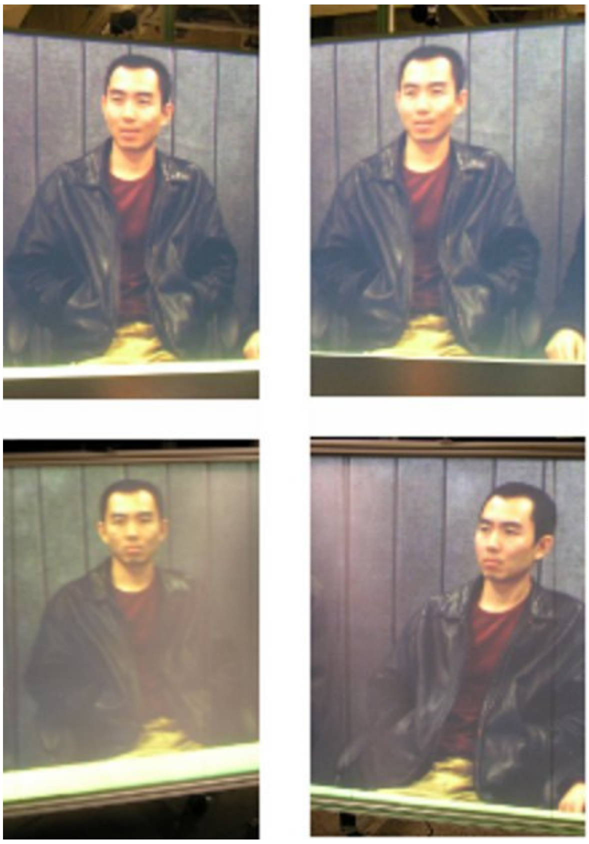

Figure 41 View of user uB from perspective of uA1 (left) and uA2 (right).
Top row shows current example, bottom row shows example session from 5.4. Adapted from [2007-Nguyen]

Different camera configurations can be used to change how the gaze error is distributed among users, but it is not possible to eliminate this error for all users, unless additional views are captured and displayed (as we discuss in the following section). As an example, the virtual space could be arranged so that the camera at site B aligns with users uA1 and uB, as shown in Figure 42. This would provide perfect eye contact for user uA1, at the cost of an increased gaze error for user uA2.

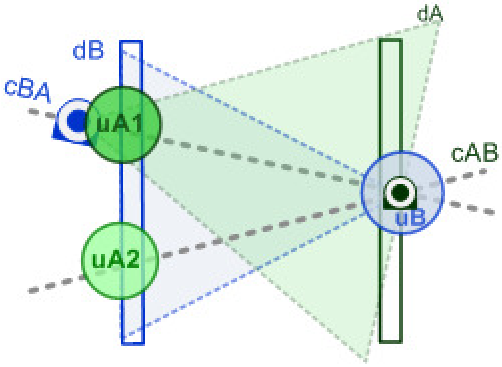

Figure 42 Alternate virtual space configuration for multi-user site (no eye contact)

The gaze errors for both session configurations are shown in Figure 43. It is interesting to note that the total gaze error in each scenario (i.e. the sum of errors for users uA1 and uA2) is the same, and equal to the angle **α** formed by the positions of uA1, uB, and uA2 in the virtual space. Moreover, the average gaze error for users at site A is also the same across configurations, and equal to **α/2.** That said, we believe that the first configuration (camera at intermediate angle between users) is the better approach, since it is more desirable to have smaller gaze errors evenly shared across users, rather than have eye contact range from great to terrible depending on the user.

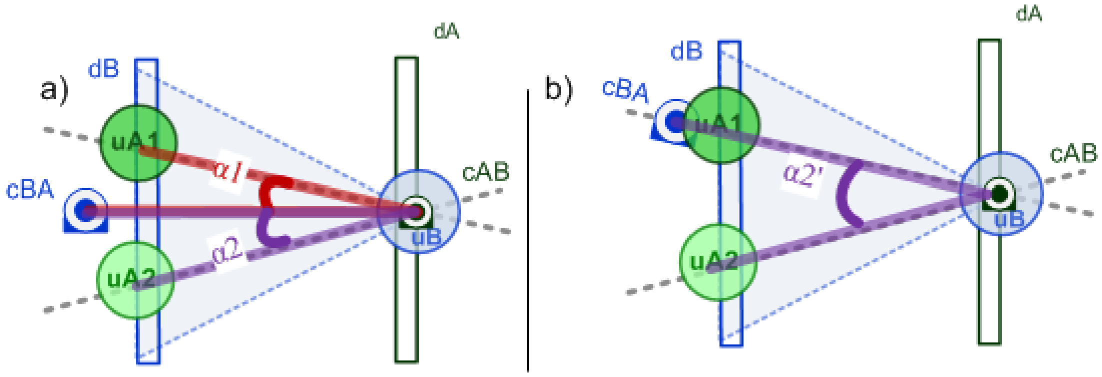

Figure 43 Gaze error, a) camera between uA1 and uA2, b) camera aligned with uA1

As we demonstrated on section 5.2.4, we can look at the geometry of a conferencing site outside the context of any session, to characterize the minimum gaze error for that site. This also applies to sites with multiple users. Following, this reasoning, Figure 44 shows site A from the current example in isolation.

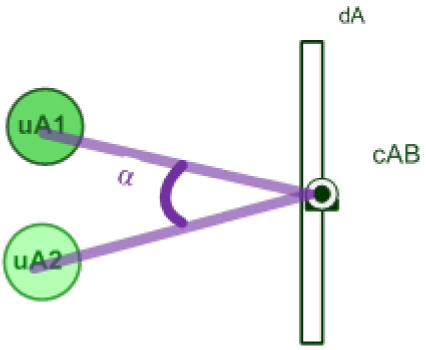

Figure 44 Minimum total gaze error, 2-user site

Assuming the location of camera cAB coincides with the position where the remote user will be rendered (which is required for optimum eye contact), the total gaze error experienced by users uA1 and uA2 will always be equal or greater than the angle **α** formed by user uA1, the position of camera cAB, and user uA2. Thus, the gaze error can be reduced by:

- Increasing the distance between display (and camera) and users.
- Decreasing the distance between users.

Both measures have their drawbacks. An increased viewing distance can reduce the perceived image quality and sense of presence, and users too close to each other can be less comfortable.

The above examples assume that the single-user site still has perfect eye contact, that is, that user uB correctly perceives gaze direction from users uA1 and uA2. This will be true if, in the common session virtual space defined for the session, the following conditions are met:

a) The line defined by the positions of users uA1, uB contains the capture position cAB
b) The line defined by the positions of users uA2, uB contains the capture position cAB
c) The rendering of user uB in display dA, **uB_dA**, is placed so that its position in the virtual space coincides with that of cAB

That is, there is eye contact at the single-user site if, at the multi-user site, the position of the camera and the displayed user image is the same. Note that this is a more strict requirement than that of the scenario with single-user sites described in 5.2. There, it was only needed that these elements were aligned, but there was flexibility to move just the camera or the display along the line (e.g. to have a camera some distance behind the display). By contrast, in this case, the position of the camera and that of the rendered user must coincide, so the camera needs to be placed in the plane of the display.

5.4 Eye contact with multiple users per site (multiple views)

In the previous section, we have demonstrated that conferencing sites with multiple users cannot provide eye contact for all local users while only displaying a single view of a remote participant. We now expand on that scenario to show how multiple captured views in combination with multi-view displays can be used to address this issue.

Consider a screen capable of displaying multiple views simultaneously, so that a user could observe a different view depending on the position of observation, as depicted in Figure 45. Such a screen could be used in a conferencing site with multiple users, in order to provide each user with a unique view of a remote site, matching the user's viewpoint.

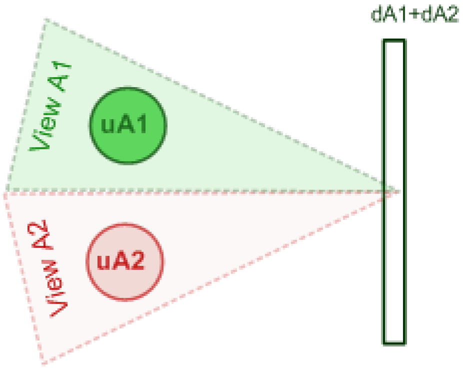

Figure 45 Multiview display

It is possible to implement a multiview screen meeting these requirements by using currently available autostereoscopic technology. Autostereoscopic displays have the ability to show different views depending on viewing position, but use it to generate a stereoscopic effect by displaying different views to each of a user's eyes. Thus, we only need to make the viewing regions larger and more spaced out to provide a single view for each user looking at the screen. A prototype screen of this kind has been successfully implemented by the 3DPresence project [2008-Schreer]. Though the 3DPresence display also supported stereoscopy in addition to multiple perspectives (which, in turn, requires several views per user), stereoscopy is not a required feature for these systems. Stereoscopy support falls outside of the scope of the telepresence model presented in this chapter, and is left as a possible extension for future work.

Figure 46 shows a session configuration that leverages a multiview display to provide eye contact at a site with two users. Like in the example sessions in section 0, there are two sites A and B, with the first site having two users **uA1** and **uA2**, and the second having a single user **uB**. The main difference is that, in this case, the display at site A, **dA1+dA2** is a multiview display that can present a different view to each user. The display at site B, **dB**, remains a conventional single-view screen. Another change from previous scenarios is the addition of a second camera at site B; that site now transmits two captured views, **cBA1** and **cBA2**, representing user uB from the point of view of users uA1 and uA2, respectively. The session is configured so that each user at site A can see the view that matches his viewpoint. At site A, there is still a single camera, capturing view **cAB**.

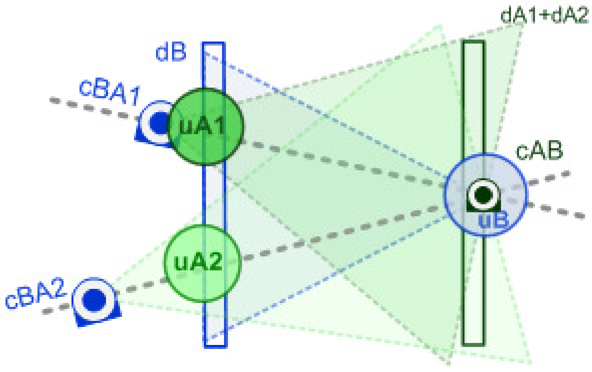

Figure 46 Virtual space of session with 2 users and 2 views displayed at site A (with eye contact)

The effect of introducing the multiview display and additional captured view for the session can be seen in Figure 47. Since each user at site A can now see a different view of the remote site, and the camera capturing that view is aligned with the viewing user and the remote user in the virtual space, the previous error in gaze direction perceived by uA1 and uA2 has been corrected. This can be explained by the fact that, at site B, whenever user uB looks straight at a remote user, the camera associated with that user is directly behind (or, more generally, aligned with) the displayed image of that user.

The eye contact provided by this configuration is also illustrated in Figure *41* (in the previous section), where it is compared with that of a session without a multiview display.

Note that the eye contact at site B, which worked properly in the previous scenarios, remains unchanged.

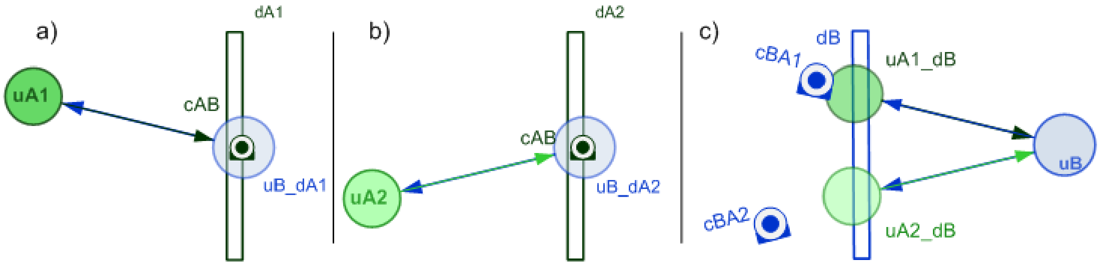

Figure 47 Eye contact for uA1 (a), uA2 (b), uB (c)

In order to provide eye contact in a session with a multiview display, the rules defined previously still apply. That is, in the common virtual space for the session, for each pair of users at remote sites, there will be eye contact between those users if their positions are aligned with their respective capture positions and the position of their displayed images.

5.4.1 Example 4: Eye contact in session with 2 sites with 2 users each

Although the examples provided in this section focus on a scenario with a single-user site and a site with 2 users, the same principles can be generalized to sessions between multiple participants, where each participating site can have any number of users. To illustrate this, consider the session represented by the virtual space in Figure 48. Here, we have two sites A and B, each with two users (uA1, uA2, uB1 and uB2), a multiview display (dA1+dA2, and dB1+dB2), and two cameras (cAB1 and cAB2, and cBA1 and cBA2). The multiview displays are configured so that each user can see the view that matches his viewpoint. All users, cameras, and displays are aligned so as to provide optimum eye contact for all users.

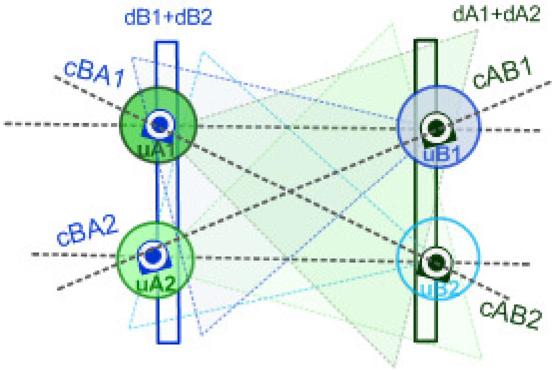

Figure 48 Virtual space for a session between two 2-user sites (with eye contact)

An important side-effect of having multiview displays at both sides of a conference is that this introduces considerable restrictions in the positioning of users, displays, and cameras. All freedom to adjust the distance between cameras and users has been lost, since the position of a camera in the virtual space needs to coincide with that of the remote user observing its captured view, and the rendered image of that user at the local site.

5.5 Spatially faithful sessions

A session with spatial faithfulness presents an optimal environment for the exchange of nonverbal cues between participating users. In such a session, all users share a common space in which the spatial relationships between objects are preserved across conferencing sites, and it is possible to know which object is being looked at or pointed at by any user, at all times.

A session is spatially faithful if:

1) A common virtual space can be defined for the session
2) At each participating site, all users perceive remote objects in the conference as matching their respective positions in the session virtual space
3) There is eye contact between all users in the session

Requirement 1) means that it is possible to define a model describing all objects in the session. An additional constraint is added by requirement 2), which states that the scenes showed to all users of the session must be consistent with this common model. Finally, requirement 3) ensures that the relative orientation of all users in the session is adequate for conveying gaze and other cues.

All the examples of sessions with eye contact shown earlier in this chapter meet these three requirements, and therefore are spatially faithful. For illustrative purposes, we show again the session from example 4 (section 5.4.1).

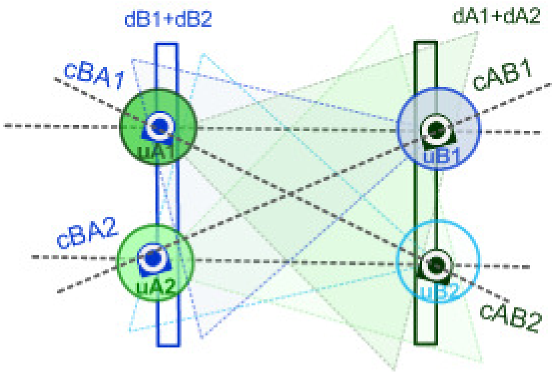

Figure 49 Spatially faithful session between two 2-user sites

We have previously established that, through the use of multiview displays, it is possible to provide eye contact for all users in this session. We can also see that the session meets the other 2 requirements for spatial faithfulness defined above: there is a common virtual space, and all users can observe scenes that match that virtual space. To verify that the session is spatially faithful, we need to determine that all users have gaze object awareness among themselves (as per the definition of spatial faithfulness, in section 5.1.1).

A user has Gaze Object Awareness with another if the user can identify the object that the other user is looking at. Since there is eye contact (also known as mutual gaze awareness), we already know that all users in this session knows when a remote user is looking at them. More generally, it is possible to know the gaze direction of a user looking at an arbitrary point. Since all objects in the session are perceived as being in the same position for all observing users, one can follow the gaze direction of a remote user and see the same object looked at by that user.

5.6 Impact of video scaling

As we discuss in section 5.1.2,usage of video scaling in a session breaks the geometry of the session space, by showing incorrect sizes and distances of remote objects. In practical terms, in a session where video scaling is applied, it is generally not possible to have a common virtual space for the session. Rather, each user can perceive a virtual space that doesn't match those of other users. Among other consequences, this makes the session not spatially faithful. In this section, we describe a technique to mitigate these negative effects of video scaling.

5.6.1 Video scaling on 3D displays

When a user observes a downscaled 2D video of a remote site, the remote objects are perceived as smaller than their real size. However, the same is not necessarily true for scaled 3D video displayed on a stereoscopic display. By properly adjusting the depth information in such a video, it is possible to have remote objects perceived as real size objects located beyond the screen. [2009-Feldmann] provides an example of using this technique to reduce display size requirements.

This is illustrated in Figure *50*. In the figure, the virtual space for a session between two sites A and B is partially represented, showing users **uA** and **uB**, the stereoscopic display of site A **dA**, and the representation of user uB at display **dA, uB_dA**. Other session elements are omitted for clarity.

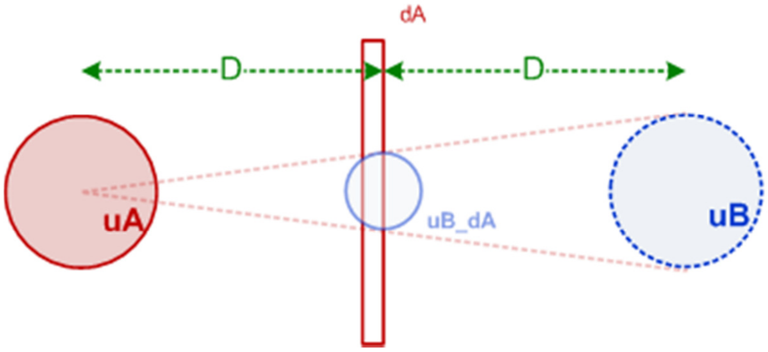

Figure 50 Video scaling and virtual distance

There is a distance **D** between user uA and the corresponding display, dA. The video of user uB in the figure is scaled to 1/2 of its real size, and the depth information for the video of user uB has been adjusted so that uB is perceived to be at a distance D behind the screen. Thus, uA perceives the remote user uB as if it were at a distance 2*D, both in terms of depth and size.

That is, by scaling a 3D video of a remote user to half its normal size and adjusting video depth accordingly, we can change the virtual distance between the local and remote users that user to double the user-display distance.a

This can be generalized as: a reduction in the dimensions of a 3D video by a scale factor of **S** translates in the session virtual space into an increase of virtual distance between the local user and the objects rendered in that video, by the same factor.

More generally, we can reduce the dimensions of a 3D video by a scale factor of **S** and increase by the same factor the virtual distance between the local user and the objects rendered in that video. If the depth information for the video matches the new virtual distance, objects in the video will be perceived as real size objects placed at a point beyond the screen. Unlike 2D video scaling, this technique can preserve object gaze awareness, and can be used in spatially faithful sessions.

Note that we refer specifically to a change of scale in video dimensions (i.e. height and width), rather than area. In the example above, only the height and width of the remote user would be halved, whereas the video area would be 1/2*1/2= 1/4 that of the original image.

5.6.2 3D Video scaling and screen size

Intuitively, scaling video to a more reduced size should allow displaying that video in smaller displays. Indeed, that is the primary motivation for using video scaling in conferencing sessions, as discussed in section 5.1.2. Under normal circumstances, downscaling a video by a certain factor **S** will reduce the screen dimensions (i.e. height and width) required to display that video by the same factor. However, in some conferencing scenarios, the need to preserve the spatial relationships between session elements makes video scaling a much less efficient method to save screen space – or even actually require *larger* screens.

Consider a session with multi-user terminals and multiview displays, like the one described in section 5.4 and Figure *45*. In such a session, and provided that video is displayed at a 1:1 scale, the dimensions of the screen required to display user uB would coincide with the height and width of the user. This is shown in Figure *51*.

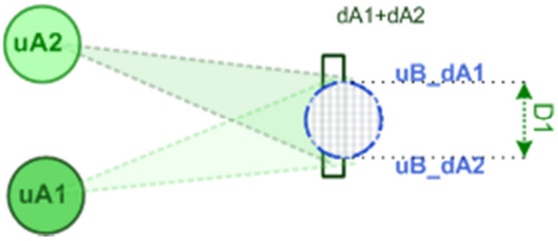

Figure 51 Screen size, multiview display

The figure shows the users at site A, **uA1** and **uA2**, the multiview display **dA1**+**dA2**, the representations of user uB in that display, **uB_dA1** and **uB_dA2**, and the minimum display width required, **D1**, which coincides with the width of user uB. Note that in this multiview display, different views of user uB are shown to uA1 and uA2 (as explained in 5.4), and that both renderings of uB are co-located and share the position of the representation of uB in the virtual space (not shown in the figure).

Now, let us illustrate the effect of video scaling on screen size for such a session by applying the scaling technique from the previous section. We want to scale the video of user uB to half its size, while preserving eye contact and spatial faithfulness. To that end, we follow these steps:

- Increase the virtual distance from uB to uA1 and uA2 by a factor of 2 (as explained in 5.6.1)
- Replace the display with a stereoscopic screen, and adjust the video depth so that uB is rendered at the new virtual distance
- Reduce the dimensions of each representation of uB to one half their original size.
- Change the rendering positions of the representations of uB, (uB_dA1 and uB_dA2) so that they no longer coincide in space.

The final result is shown in Figure *52*:

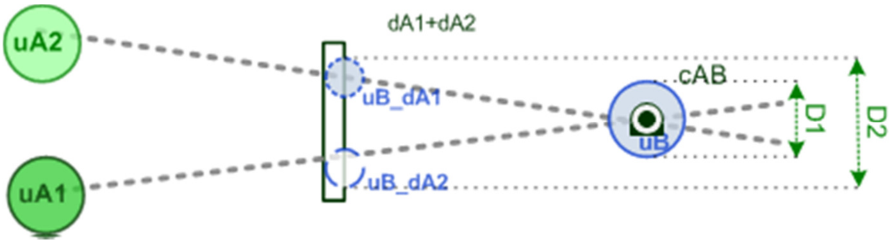

Figure 52 Screen size, multiview display with scaled video

Note that, in order to preserve eye contact between users, the renderings of user uB (uB_dA1 and uB_dA2) no longer share the same position, but they need to be aligned with uB, the observing user, and the camera at site A (which is co-located with uB in the virtual space). This has a negative impact in the required display size, since both representations of uB need to fit into the multiview display. The minimum display width required for this scenario, **D2**, is shown in the figure, alongside the original required width, **D1**. As we can see, the required screen width has actually increased by downscaling the video.

Although there are workarounds to this effect (such as using multiple smaller displays at specific locations, rather than a larger one), we should consider that, in general, for sessions with multiview displays, it may not be possible to reduce screen size requirements by scaling video while preserving eye contact and spatial faithfulness.

5.7 Generating a session virtual space

The generation of a virtual space for a multimedia session is a crucial step in the methodology for initiating a multiview video session described in chapter 0. However, it is also an open problem with a large variety of possible solutions. The purpose of this section is to provide an overview of the challenges involved in the process, and to suggest a series of guidelines that can be followed to address these challenges.

5.7.1 Problem Overview

In order to define a virtual space for a session, a session focus should take the following steps:

- Obtain a list of user agents participating in the session, and the capabilities and site geometry associated with each participant. This is discussed in chapter 4.
- Determine which session features are supported by this set of participants, and decide a feature set for the session
- Generate a virtual space that meets the required features and is compatible with participant capabilities and topology.

Certain specific scenarios may further complicate this process:

- The list of participants might not be static, but change over time. When a new user agent joins the session, should the session focus try to fit it in the existing virtual space, or discard that space and create a new one? Should session features be re-evaluated? What about when a participant leaves the session?
- Likewise, the geometry or capabilities of participating user agents can change, even if no participants join or leave the session: users entering or leaving a conferencing site, network conditions changing, or objects in a conferencing site changing position are some examples of this. Again, the session focus could adapt the existing space to the new conditions, or create a new one.
- In some cases, it may not be possible to generate a single virtual space suitable for all session participants. In this case, the compromise solution is to define multiple spaces, each associated to one or more participants. This makes it impossible to provide the spatial faithfulness feature, but it still allows to offer eye contact, with an appropriate configuration. In extreme scenarios in which terminals have multiview displays, it is even possible that multiple virtual spaces per participating user agent will be needed – in these cases, each space is associated to an individual user, rather than a user agent.

In the following sections, we discuss the basic scenario, leaving some of the additional considerations explained above as a future work.

5.7.2 Determining session features

The following features of a multimedia session interact with the configuration of a session virtual space, either by depending on it, or by imposing restrictions on how virtual space elements can be arranged.

Eye contact. The quality of eye contact depends on the proper alignment of users, cameras, and screens on the virtual space, as discussed in sections 5.2, 5.3, and 5.4. It is also limited by the intrinsic properties of each participating terminal, as explained in 5.2.4. In order to have eye contact between all users in a session, all participating terminals must support the feature, and it must be possible to define a virtual space where session elements are correctly aligned. Alternatively, when this is not possible, a session can be configured to have eye contact only between a subset of its users.

A conferencing terminal is usually limited in the number of users for which it can provide eye contact. In addition to the requirements stated in previous sections, the following considerations apply:

- At any terminal, up to N remote participating users can have eye contact with users in that terminal, with N is the number of capture devices that can be aligned with local displays. Note that this only affects eye contact as perceived by the remote users; the number of local cameras has no impact in the eye contact observed by local users.
- At a multi-user terminal, up to M local users can have eye contact with remote users, where M is the number of views provided by multiview displays in that terminal. If no multiview displays are available (or if not enough views are provide), some local users will perceive eye contact information corresponding to other users – they will believe being looked at when they are not, and vice versa.

Spatial Faithfulness. The requirements for a spatially faithful session are discussed in detail in 5.5. To summarize, the session must provide eye contact to all users (as discussed above), to have a single common virtual space shared among all participants, and to have the rendered scenes at each site match the geometry of this virtual space.

In order for this to be possible, the geometries of participating sites need to meet certain specific properties:

- Virtual distance. In order for the perceived distances between users to remain consistent across sites, the user-display distances should be the same at every for every terminal. Alternatively, if stereo video is available, depth information combined with video scaling can be used to adjust virtual distances between users to arbitrary values, while preserving the perception of “real size” video, as explained in section 5.6.
- Real size. Strict spatial faithfulness requires video to be rendered at real size or, alternatively, to be rendered in 3D with a depth value that matches the size observed by users, as explained in section 5.6.
- Viewing angle. When more than two participating sites are present in a session, they are usually distributed uniformly around a circular table. In a spatially faithful session, this requires the screens at each site to cover a wide angle around the users, and this width increases with the number of participants: a three-participant session requires more than 60º of separation between participants, a four-participant session requires angles above 90º, and so on. This requirement is illustrated in Figure *53*, which shows the virtual space of two multi-participant sessions with a local user (in red) and two or three remote users, each associated with an individual display. For simplicity’s sake, in this figure and subsequent ones in these sections, only a relevant subset of virtual space elements is shown.

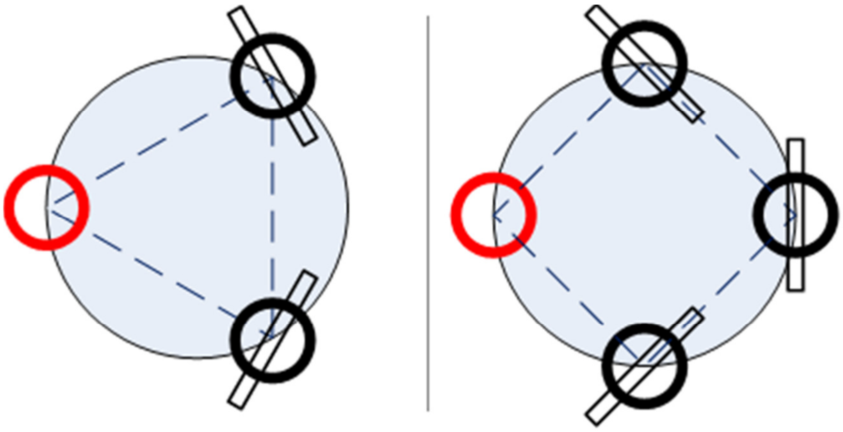

Figure 53 Participant distribution in spatially faithful session

Scale and Framing. The size and number of screens in a terminal determines the number of remote users that can be shown in real size (if any) at a given session. In section 5.1.2, we suggested as a guideline that a full size user with upper body framing takes a frame of about 50x70cm. Other factors in the virtual space can limit the use of real size video or otherwise affect scale and framing, including user distribution in the space, or the need to preserve eye contact.

5.7.3 Generation of a virtual space for a set of participant capabilities and session features

The process for generating a virtual space varies depending on the features of the session. Here we address some of the most common scenarios.

5.7.3.1 Spatially faithful session, single-user sites

When the session is spatially faithful, and all participating sites have a single user each, a good solution is to arrange all users at regular angular intervals around a circular table, as shown in Figure *53*. Here, the angle of separation is determined by the number of participants, and the radius of the table is based on the distance between users and displays. This imposes very strict requirements on the physical disposition on elements at a conferencing site, making difficult for any terminal to be able to handle sessions with different numbers of participants, or different table sizes.

As we mentioned in previous sections, a way to introduce some flexibility in the configuration of spatially faithful sessions is to have 3D screens and replace real size video with scaled 3D video with appropriate depth values. If done right, this should provide the same level of immersion as using real size video. This is illustrated in Figure *54*, which shows several configurations for spatially faithful sessions that are possible for a terminal with a 3D display. The local user and local 3D display are represented by a red circle and a green rectangle, respectively, whereas remote users are black circles arranged around a circular virtual table. From left to right, the configurations correspond to a 3-participant session, another 3-participant session with larger inter-user distances, and a 4-participant session.

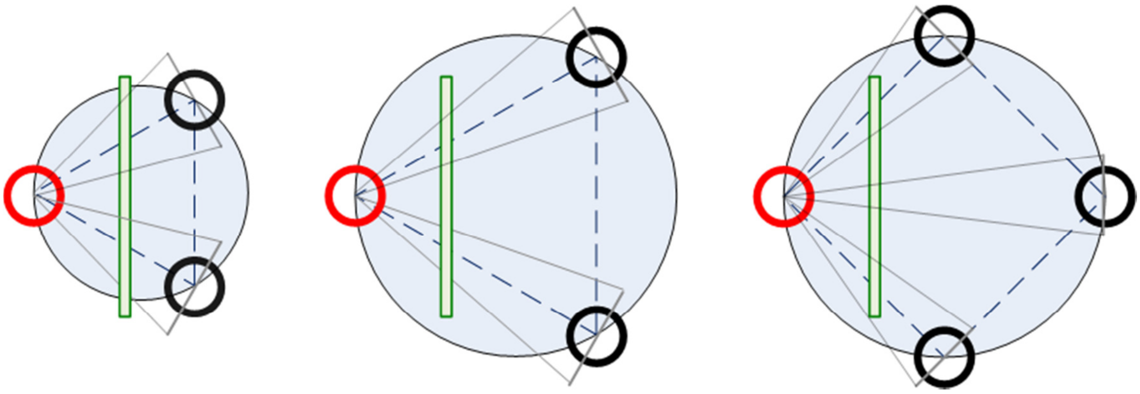

Figure 54 Spatially faithful session configurations with 3D display

Another consideration that must be taken into account when generating a virtual space for a spatially faithful session is to preserve eye contact – user positions must be aligned with displays and cameras, as usual.

5.7.3.2 Spatially faithful session, multi-user sites

When a spatially faithful session has multiple users per participating site, the virtual space generation process is very similar to the one described in the previous section. The main difference lies in how users are arranged around the virtual table: rather than distribute uniformly all users, it is better to have one group of users per participating site, and arrange the groups at regular intervals around the table. This simplifies the configuration, and allows some flexibility in the physical distance between users at the same site, or even in the number of users that each site has. Figure 55 illustrates this principle, by showing a 3-participant session where each site has 2 users.

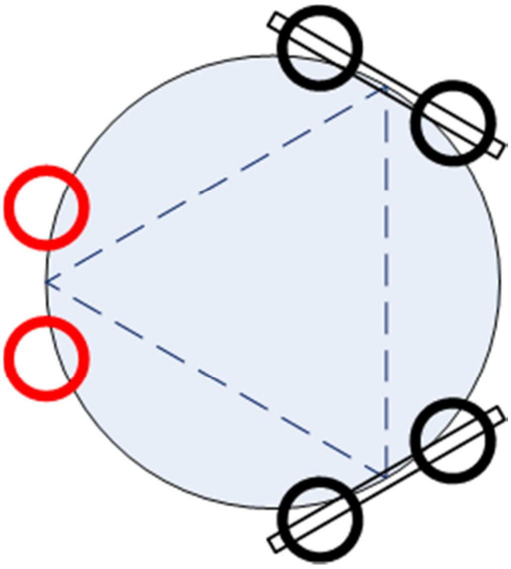

Figure 55 Spatially faithful session with 3 participants and 2 users each

Note that the eye contact requirements for these sessions imply that any conferencing site with 2 or more participants needs to have multiview displays with that many views.

5.7.3.3 Session with eye contact, single-user sites

Compared to those of spatially faithful sessions, the virtual spaces of sessions without spatial faithfulness have a huge flexibility in how they can be configured. For sessions with eye contact, the only real restriction is that the position of remote users must be aligned with a capture device (or with a virtual view).

When we consider that it is now possible to have a separate virtual space for each participating site, the possibilities expand even further. This is illustrated in Figure 56. There, a session with three participating sites A, B, and C is shown, each with a single user (uA, uB, and uC, respectively), two displays, and two cameras. In the figure, two different virtual spaces are shown. On the left side, the virtual space for site A has participants regularly distributed around the table like in previous examples; also, each remote user is rendered at a position aligned with a local camera.

However, the virtual space for site B (at the right) has a different arrangement, with users uA and uB clustered together to fit in the local displays, which are much closer. The original positions of uA and uB in the virtual space associated with site A are shown, faded, next to those of the virtual space of site B, for comparison. Though this configuration is clearly not spatially faithful, it still provides good eye contact, since cameras are aligned with user representations at both site A and site C. The virtual space for site B is not shown, but it could be a copy of either the one in A or in C, or a different one altogether.

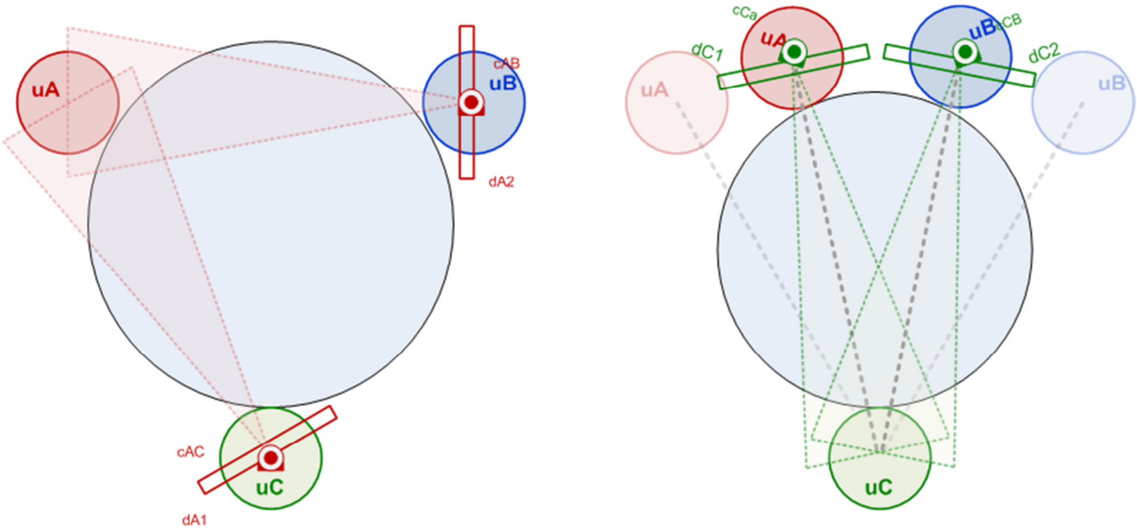

Figure 56 Session with eye contact and multiple virtual spaces

Given the degree of freedom of virtual space generation in this kind of sessions, it is difficult to provide specific guidelines as to how to proceed. As an example, the conference focus could want to emphasize any of the following factors in a conference:

- Quasi-spatial faithfulness. When full spatial faithfulness is not feasible, it may be desirable to have a configuration as close as possible to it. Such a session would have virtual spaces for each site that are different but still very similar to one another, and part of the benefits of spatial faithfulness, such as being able to tell who other users are talking to or signalling, with some degree of accuracy.

- Best scale. Alternatively, a configuration might completely dismiss spatial faithfulness and go for the arrangement that better leverages the available display areas. In this case, users would be shown in as large a scale as possible, possibly at the cost of spatial inconsistencies, though still preserving eye contact.
- Full screen, selective transmission. An extension of the previous point, in this case the desire for better video scale could go as far as showing only a subset of participating remote users at any given time, leaving the rest in voice only, or in small windows, and changing the active user based on floor control mechanisms. In this case, eye contact could also be preserved, but only with the active remote users.

5.7.3.4 Session with eye contact, multi-user sites

When the session to configure has eye contact but not spatial faithfulness, the suggestions from the previous section apply. Again, there is a great flexibility in the range of solutions that can be applied. The main difference with the previous scenario is the need for conferencing sites to have multiview displays showing several views in order to provide eye contact for multiple users.

5.7.3.5 Session with no eye contact

If a session does not require eye contact, its virtual spaces can have a completely arbitrary configuration, as long as they can be rendered within their corresponding conferencing sites. As a matter of fact, these sessions don’t usually require a virtual space at all, and can be configured without need of one.

6 Conclusions and Future Work

Multiview video technology extends the functionality of a traditional video stream to provide new features, like depth information in stereoscopic video, adjustable video perspectives in free viewpoint video, or shared virtual environments with perfect eye contact in immersive conferences.

If one wanted to deploy a communication system with multiview video today, the technology on the media side is fairly mature, with abundant standard support and commercial hardware available for encoding, display, capture, and other parts of the media processing chain. The same cannot be said about the network plane: current communication protocols are unaware of multiview video technologies, and it is not possible to configure a multiview video session using existing standard specifications. This thesis proposes a solution for this issue.

The main objective of the thesis has been to **extend the Session Initiation Protocol (SIP) to support the negotiation of multimedia sessions with multiview video streams**. This has been achieved through three primary contributions: a SIP extension for 3D video, a SIP extension for multiview video conferences, and a model for the virtual space of a session.

Extension to SIP/SDP for the signalling of stereoscopic video streams

The first step in the development of our signalling solution for sessions with stereoscopic video has been to identify the requirements of 3D video streams from a signalling standpoint. Our analysis revealed two major points that needed addressing: the selection of a 3D video format for the stereoscopic stream, and a way to describe the relationship between the multiple media streams that can compose a 3D video stream.

In order to meet these requirements, we have specified an extension to the Session Description Protocol (SDP) [RFC4566], which is the document format used to describe and negotiate the configuration of media streams in SIP sessions. The extension consists on a new media-level attribute for SDP, called “3dvFormat”, and a new type of decoding dependency for SDP, called “3dd”.

The “3dvFormat” attribute can be used to associate a 3D video format with a 3D video stream. Our specification provides support for four 3D video formats: simulcast stereo video, simulcast video and depth, video-plus-depth, and frame-packing. Note that another common 3D video format (Multiview Video Coding) has been deliberately left out of the specification, due to the fact that it is covered by a future standard, currently in progress [ID-MVC-RTP].

The “3dd” decoding dependency is used to describe the relationship between different media streams composing a 3D video stream. It is based on the framework for signalling decoding dependency in SDP [RFC5583], which allows the definition of hierarchies between streams, and provides a flexible way to negotiate different 3D format configurations.

Extension to SIP for the signalling of multiview video conferences

For our solution for the signalling of conferences with multiview video, we also started with a detailed requirement analysis. In this case, three main requirements were identified. We needed a way for SIP user agents to exchange information about their topology and multiview capabilities, a method for the focus of a conference to define a virtual space for the conference and a map of media streams and send this information to user agents, and a signalling process that integrated the previous two steps with the SIP invitations used to initiate a conference.

To address these requirements, we have specified an extension to SIP, using the framework for specific event notification [RFC3265] to define two new event packages. Event packages use the SIP subscribe/notify mechanism to exchange information between SIP entities, usually in the form of XML documents. Our first event package is called “mvv-info”, and describes the multiview video information associated with a user agent; the second event package, “mvv-conf-info” is used for the configuration information for a multiview video conference that cannot be conveyed through the usual mechanisms in SIP (i.e. cannot be included in SDP). Finally, we have provided a detailed signalling process that has user agents and conference focus exchange the information for these event packages prior to the sending of a SIP INVITE request to start the conference.

The event package “mvv-info”, for multiview video information, provides a description of some properties of a SIP user agent that aren’t included in a typical SIP negotiation process, such as terminal topology or multiview capabilities. By terminal topology, we refer to the spatial distribution of objects in a conferencing site, such as terminals, displays, and users. The multiview capabilities include the number of supported media streams, as well as other parameters like available bandwidth or supported media formats.

The event package “mvv-conf-info”, for multiview video conference information, describes certain aspects of conference configuration that aren’t present in the regular SIP negotiation process, like conference virtual space and stream map. The virtual space of a conference is a model that assigns a position in space to the elements of each conferencing site participating in a conference, including terminals, displays, and users. A stream map is a list of all the media streams that are to be exchanged during a conference, with information like the origin and destination of each stream.

The signalling sequence that has been defined for multiview video sessions has user agents send their “mvv-info” documents to the conference focus in the first place, followed by the conference focus defining and sending a “mvv-conf-info”. Once all conference participants have this information, the conference can be started with a SIP INVITE request, as per normal SIP operation. These additional signalling exchanges are justified by the rich configuration information that is required to setup a conference with multiview video.

A model for conference virtual spaces

As part of our extension for multiview video conferences, summarized above, we have specified a model for the virtual space of a conference. The virtual space defines a coordinate system for the conference, and associates each user, camera, and display at a participating terminal with a position in that coordinate system. These positions are assigned by the focus of the conference during the configuration process, and are ideally shared by all participating user agents, so that each user in the conference perceives the same scene. The disposition of a virtual space is deeply related with certain features that can be provided by a conference, like eye contact, spatial faithfulness, or video scale. Thus, we have defined mechanisms to analyze a virtual space in order to determine the features it supports and, conversely, we have provided guidelines to generate a virtual space that can support a given set of features.

6.1 Future Work

To continue the research initiated with this thesis, we have identified two promising lines of future work: to pursue the standardization of our proposed protocol extensions, and to work in the implementation of a conferencing system based on the framework we have defined.

Standardization

Though the extensions to the Session Initiation Protocol that are proposed in this thesis are valuable as a purely theoretical exercise, or as a solution for the development of proprietary systems, the only way to truly address the underlying problem (i.e. lack of a common signalling for multiview video sessions) is to define a standard to that end. Indeed, our aim when writing this document was to provide specifications that could eventually be turned into standards – or, at the very least, that could serve as starting points in a standardization process. That said, this may be easier to achieve for some parts of the thesis than others.

At the point of writing this, we are actively participating in standardization activities regarding one of the major contributions of this work: the SIP extension for 3D video. The organization working in this topic is the Internet Engineering Task Force [IETF], and specifically the working group for Multiparty Multimedia Session Control (mmusic) [MMUSIC]. This working group has set a milestone for submitting "SDP extension to signal stereoscopic 3D video as Proposed Standard", and to this goal we have been providing feedback and suggestions, which culminated in the submission of the body of chapter 3 of this thesis as a proposed Internet Draft.

As for the SIP extension for multiview video conferences, no immediate action has been taken towards standardization, due to the fact that there is some overlap with some ongoing standardization efforts. To be more specific, the multiview video conferences covered by our specification have many common requirements with telepresence sessions.

Signalling for telepresence systems is not fully standardized yet, but this is being worked at by the IETF working group for Controlling Multiple Streams for Telepresence (CLUE) [CLUE]. The CLUE specifications are still at an early stage, and relatively unstable, since the working group is very active, but we can tell that they are trying to solve some of the same problems as our multiview extension. For this reason, we consider it unlikely that the contents of chapter 4 will be adopted by the IETF as is, but we still expect that they can be the basis for productive discussion, and maybe a revised version can be submitted at some point in the future.

System Implementation

We believe that integrating the ideas presented in this thesis into a working conferencing prototype would be a productive line of research, as well as a way to further validate our work. It must be noted, though, that the proposed signalling extensions and models are not strictly theoretical constructs, since early versions of them were implemented and tried for the project CENIT-VISION, as described in [2011-Perez]. However, the signalling framework has been considerably expanded and improved since the conclusion of VISION, to the point that trying it on a new system could be very useful.

That said, the kind of system that could be used to fully validate this work is not an easy one to accomplish. Stereoscopic video by itself can be implemented with relatively low hardware and personnel requirements, but the more advanced features of a multiview conference are a different matter. In order to achieve free viewpoint video and shared virtual environments, CENIT-VISION built dedicated conferencing rooms with dozens of PCs working in parallel, and had a large consortium of enterprises and universities working over four years on the subject. Our hypothetical conferencing system would probably require a similar amount of resources.

7 References

[2000-Kang] Kang, S.B., Szeliski, R., Anandan, P. “The Geometry-Image Representation Tradeoff for Rendering”Proceedings 2000 International Conference on Image Processing, vol. 2, pp 13—16 (2000)

[2002-Chen] Chen, M. “Leveraging the Asymmetric Sensitivity of Eye Contact for Videoconference”. Proc. CHI 2002 (2002)

[2002-Monk] – Monk, A., Gale, C. “A look is worth a thousand words: full gaze awareness in video-mediated Conversation” Discourse Processes (2002)

[2002-Schreer] O. Schreer, P. Kauff, “An Immersive 3D Video-Conferencing System Using Shared Virtual Team User Environments”, Proc. of ACM Collaborative Virtual Environments (CVE 2002), (2002)

[2003-Baker] Baker, H. et al. “Computation and performance issues In coliseum: an immersive videoconferencing system”, 11th ACM International Conference on Multimedia (2003).

[2003-Gross] – Gross, M. et al. “Blue-c: A Spatially Immersive Display and 3D Video Portal for Telepresence”, ACM Transactions on Graphics (2003)

[2003-Vertegaal] Vertegaal, Weevers, Sohn, Cheung. “GAZE-2 Conveying Eye Contact in Group Video Conferencing Using Eye-Controlled Camera Direction”, Proc. CHI 2003, ACM Press (2003)

[2005-Nguyen] Nguyen, N., Canny, J. “MultiView: Spatially Faithful Group Video Conferencing” Proceedings of the SIGCHI conference on Human factors in computing systems (2005)

[2007-Alatan] Alatan, A.A., Yemez, Y., Güdükbay, U., Zabulis, X., Müller, K., Erdem, C. E., Weigel, C., Smolic, A. “Scene Representation Technologies for 3DTV—A Survey.” IEEE Transactions on Circuits and Systems for Video Technology,vol. 17, No. 11 (2007)

[2007-Gamer]Gamer, M., Hecht, H. “Are you looking at me? Measuring the cone of gaze”. Journal of Experimental Psychology: Human Perception and Performance (2007)

[2007-Konrad] J. Konrad and M. Halle, “3-D displays and signal processing,” IEEE Signal Processing Mag., vol. 24, no. 7, pp. 97–111 (2007).

[2007-Kubota] Kubota, A. Smolic, A., Magnor, M., Tanimoto, M., Chen, T., Zhang, C.” Multiview Imaging and 3DTV”. IEEE Signal Processing Magazine, vol. 1053, (2007)

[2007-Muller] K. Müller, P. Merkle, and T. Wiegand, “Compressing time-varying visual content,” IEEE Signal Processing Mag., vol. 24, no. 7, pp. 58–67, (2007).

[2007-Nguyen]- Nguyen, N., Canny, J. “MultiView: Improving Trust in Group Video Conferencing Through Spatial Faithfulness”. Proceedings of the 2007 ACM Conference on Human Factors in Computing Systems (2007).

[2007-Smolic] A. Smolic, K. Müeller, N. Stefanoski, J. Ostermann, A. Gotchev, G.B. Akar, G. Triantafyllidis, A. Koz, "Coding algorithms for 3DTV—A survey," IEEE Trans. Circuits Syst. Video Technol., (Special Issue 3DTV MVC) (2007).

[2008-Schreer] Schreeer, O., Feldmann, I., Atzpadin, N., Eisert, P., Kauff, P., Belt, H.J.W. "3DPRESENCE – A System Concept for Multi-User and Multi-Party Immersive 3D Videoconferencing." 5th European Conference on Visual Media Production, pp 1--8 (2008)

[2009-Feldmann] Feldmann, I., Schreer, O., Kauff, P. Schafer, R., Fei, Z., Belt, H., Divorra, O. "Immersive Multi-User 3D Video Communication" Proc. of Int. Broadcast Conference (IBC 2009), (2009)

[2009-Jones] Jones, A. et al. "Achieving Eye Contact in a One-to-Many 3D Video Teleconferencing System" ACM Trans. Graph (2009)

[2009-Nguyen] Nguyen, N., Canny, J. "More than Face-to-Face: Empathy Effects of Video Framing". Proceedings of the 2009 ACM Conference on Human Factors in Computing Systems (2009)

[2009-Smolic] A. Smolic ,K. Mueller, P. Merkle, A. Vetro, "Development of a new MPEG Standard for Advanced 3D Video Applications" ,in:ISPA2009,6th International Symposium on Image and Signal Processing and Analysis, Salzburg, Austria,(2009).

[2009-VanderPol] van der Pol, D. "The effect of slant on the perception of eye contact," Master's thesis, Eindhoven University of Technology, 2009.

[2010-Van Eijk] van Eijk, R. "Human sensitivity to eye contact in 2d and 3d videoconferencing". International Workshop on Quality of Multimedia Experience (QoMEX) (2010)

[2010-Vetro] A. Vetro. "Frame Compatible Formats for 3D video distribution", EEE International Conference on Image Processing (2010)

[2011-Hecth] Hecht, H. " The cone of Gaze " 4th International Conference on Human System Interactions (2011)

[2011-Perez] Pérez, L. et al. "Network convergence and QoS for future multimedia services in the VISION project", Computer Networks (2011)

[2012-Moubayed] Al Moubayed, S., Edlund, J., Beskow, J., "Taming Mona Lisa: Communicating Gaze Faithfully in 2D and 3D Facial Projections". ACM Transactions on Interactive Intelligent Systems (2012)

[CENIT-VISION] Project CENIT-VISION. http://vision.tid.es

[CLUE] IETF, ControLling mUltiple streams for tElepresence (clue) working group. http://www.ietf.org/dyn/wg/charter/clue-charter

[H264] ITU-T and ISO/IEC JTC 1, “Advanced video coding for generic audiovisual services”, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), (2010)

[H323] “Packet-based multimedia communications systems”, ITU-T, H.323 (2009)

[HDMIv1.4a] HDMI, “HDMI Specification Version 1.4a” (2010)

[ID-RTP-MVC] Wang, Y., Schierl, T., Skupin, R. “RTP Payload Format for MVC Video”. Internet Draft (2011) – (Work in progress)

[IETF] The Internet Engineering Task Force (IETF). http://www.ietf.org/

[ISO/IEC 23002-3] ISO/IEC JTC1/SC29/WG11, “ISO/IEC FDIS 23002-3 Representation of Auxiliary Video and Supplemental Information”, Doc. N8768, Marrakech, Morocco, January 2007.

[ITU-T] ITU Telecommunication Standardization Sector. http://www.itu.int/

[ITU-T H.264] ITU-T and ISO/IEC JTC 1, “Advanced video coding for generic audiovisual services”, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), (2010)

[MMUSIC] IETF, Multiparty Multimedia Session Control (mmusic) working group. http://www.ietf.org/dyn/wg/charter/mmusic-charter

[MPEG-C-PART3] ISO/IEC JTC1/SC29/WG11, “ISO/IEC FDIS 23002-3 Representation of Auxiliary Video and Supplemental Information”, Doc. N8768, Marrakech, Morocco, (2007)

[MVC] Annex H of ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services", (2010)

[MVC-RTP]

[RFC2119] Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, (1997).

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E. “SIP: Session Initiation Protocol”. RFC Editor, RFC 3261 (2002)

[RFC3264] Rosenberg, J. and H. Schulzrinne, “An Offer/Answer Model with Session Description Protocol (SDP)”, RFC 3264, (2002).

[RFC3265] Roach, A. “Session Initiation Protocol (SIP) – Specific Event Notification”. RFC 3265 (2002)

[RFC3515] Sparks, R. “The Session Initiation Protocol (SIP) Refer Method”. RFC 3515 (2003)

[RFC3550] Schulzrinne, H. et al. “RTP: A Transport Protocol for Real-Time Applications” IRFC 3550 (2003)

[RFC4353] Rosenberg, J. “A Framework for Conferencing with the Session Initiation Protocol (SIP)”. RFC Editor, RFC 4353 (2006)

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, “SDP: Session Description Protocol”, RFC 4566, (2006).

[RFC4733] Schulzrinne, H., Taylor, T. “RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals” RFC Editor, RFC 4733 (2006)

[RFC5226] Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs”, BCP 26, RFC 5226, (2008).

[RFC5234] Crocker, D., Ed., and P. Overell, “Augmented BNF for Syntax Specifications: ABNF”, STD 68, RFC 5234, (2008).

[RFC5583] Schierl, T. and S. Wenger, “Signaling Media Decoding Dependency in the Session Description Protocol (SDP)”, RFC 5583, (2009).

[RFC5888] Camarillo, G. and H. Schulzrinne, “The Session Description Protocol (SDP) Grouping Framework”, RFC 5888, (2010).

[XML-NS] Bray, T. et al. "Namespaces in XML", W3C recommendation: xml-names, (1999)