A 3D VIDEOCONFERENCING SYSTEM WITH 2D BACKWARDS COMPATIBILITY

J.L. Landabaso T. Mansi C. Molina, K. Zangelin, P. Enfedaque, J. Cañadas, L. Lizcano

Telefónica Investigación y Desarrollo (TID)
Multimedia Applications Division
Via Augusta, 177 08021, Barcelona, Spain
{jlldiaz,tmansi,cmdr,kzangeli,pev,jcr,lle}@tid.es

ABSTRACT

This paper presents a 3D videoconferencing system that is 2D backwards compatible. The system offers an inmersive experience to users with 3D displays at their disposal, but does not require the users to employ dedicated hardware at the sender side. Backwards compatibility is achieved by transmitting 3D content in the user data fields of the video streams, which older teleconference terminals ignore. Other highlights of the system include the use of a conventional inexpensive camera at the sender side, low bandwidth overhead and low CPU usage. The system proposed is a necessary step towards more complex systems that will tackle the 3D videoconferencing problem as a whole. Our focus is to make it possible to reuse existing capture devices to create 3D content that will be consumed by users with specialized hardware. As 3D videoconferencing technology evolves, more professional stereoscopic cameras will follow as a natural consequence.

***Index Terms*—** 3D, Inmersive, videoconference, teleconference, foreground segmentation, H.263

1. INTRODUCTION

Videoconferencing has been one of the fields where video coding and networking technologies have had a major impact, driving successful deployment of several systems used worldwide.

Inmersive extensions to classic videoconferencing systems have been proposed in the past, but they usually require both the sender and receiver sides to purchase expensive equipments preventing a broad deployment of this technology.

Among the state of the art proposals for inmersive videoconferencing, there have been several different approaches. Some of them try to achieve a *to be there* experience by using one or multiple large 2D displays [1, 2] at the receiver side. Other systems try to provide an inmersive feeling through shutter glasses [3, 4] or similar technologies. Moreover, some of these more advanced systems include tracking modules at the receiver side that let the display render the correct images depending on the position of the observer.

Our goal is to help bringing a 3D experience to the average user of traditional videoconferencing systems. To do so, we have focused on developing an alternative approach that is able to extract 3D content from a standard commodity capture device so that 3D-ready receivers will still be able to create a closer sensation of teleinmersion. We argue that the key factor that is preventing a mass adoption of 3D videoconferencing systems, and especially auto-stereoscopic displays, is the lack of inexpensive 3D capture devices, and therefore we believe that some more efforts have to be put into facilitating an online 3D extraction process. It is important to remark that in our proposal any user will produce 3D information employing a conventional WebCam and consuming few additional resources.

3D information can be obtained using either volumetric reconstruction techniques such as [5, 6, 7] or by extracting the depth information by using stereo correspondence algorithms. A good literature review on stereo correspondence algorithm has been done in [8]. Our proposal is related to the later one. In this context, we propose to use two levels of depth only. The first one corresponds to the depth of the teleconferee, and the second one to the depth of the whole background scene. In fact, the binary depth map that we propose resembles very closely the real depth map in a videoconferencing situation.

In order to obtain a good segmentation of the teleconferee we propose a fast yet effective foreground Bayesian segmentation scheme that overcomes some of the problems of traditional exception-to-background systems. Codification of the depth data is solved using the user data fields of the video streams, enabling backwards compatibility with current systems. And finally, a non-intrusive 3D display is used on the receiver side.

The paper is organized as follows. Section 2 presents an overview of the system. Section 3 describes the techniques for pixel-domain analysis leading to the segmented foreground object blobs. This section also discusses the issues concerning color and texture-based shadow detection. In section 4 our proposal for video + depth coding is presented. Section 5 describes the image synthesis and display employed at the receiver side. Section 6 gives some experimental results and the paper concludes in Section 7.

2. SYSTEM OVERVIEW

Figure 1 shows a detailed overview of the proposed system. There are several modules split into the sender and the receiver parts of the system. The receiver side comprises the foreground segmentation module and the video + depth coder. In the receiver side, a decoder extracts the video and depth information which are displayed by means of a 3D display. Conventional receiver terminals decode the video information and disregard the rest for not being able to interpret it. All these modules are detailed in the following sections.

3. 2D FOREGROUND SEGMENTATION

Over the years many works have been published on the two dimensional foreground segmentation task, describing different methods that treat to extract that part of the scene containing active entities. In most of the cases, the stochastic background process is modeled first, and then the foreground pixels are classified as an exception

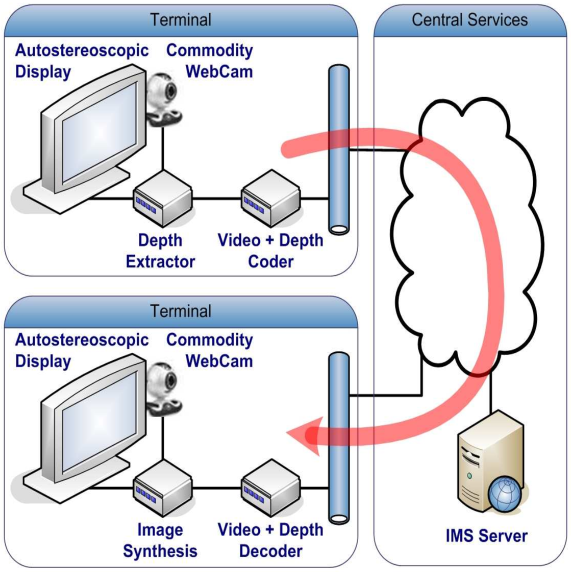


Fig. 1. The system block diagram showing the chain of functional modules.

to the model. In this paper we propose using a simple yet effective foreground model in a Bayesian classification framework which outperforms exception-to-background settings.

3.1. Single-Class Adaptive Background Models

We adopt a single-class statistical model for modeling the background color of a pixel $\mathbf{x}$ (indicating its spatial coordinates), given observations of its color value $\mathbf{I}(\mathbf{x})$ across time. For this purpose, we use a Gaussian probability density function. Gaussians have been previously proposed in [9], among others, to ensure that the cameras thermal noise does not produce classification errors. Some of these works adopt multi-class models to model repetitive background, such as in waving flags, or moving trees. However, a single-class model is enough in our approach since our system is being developed to operate in a scene that consists of a relatively static situation:

$$G_{\mathbf{x}}(\mathbf{I}(\mathbf{x})) = \frac{1}{(2\pi)^{3/2}\sqrt{|\mathbf{\Sigma_x}|}} e^{-\frac{1}{2}(\mathbf{I}(\mathbf{x})-\mu_{\mathbf{x}})^T \mathbf{\Sigma_x^{-1}}(\mathbf{I}(\mathbf{x})-\mu_{\mathbf{x}})}, \quad (1)$$

corresponding to the Gaussian that models the color of the background process of pixel $\mathbf{x}$, and where pixel color values ($\mathbf{I}(\mathbf{x})$) are expressed as a vector of three dimensions in the RGB color space. It is often assumed that the covariance matrix is diagonal with R, G and B sharing the same variances: $\mathbf{\Sigma_x} = \sigma^2_{\mathbf{x}} \cdot \mathbf{Id_{3\times 3}}$.

Similarly as in [9], model adaptation is implemented as a low pass filter procedure. Thus, once the pixel value has been classified into the background, the model is adapted as follows

$$\begin{aligned} \mu_{\mathbf{x}}[t] =& (1-\rho)\mu_{\mathbf{x}}[t-1] + \rho\mathbf{I}(\mathbf{x}) \\ \sigma^2_{\mathbf{x}}[t] =& (1-\rho)\sigma^2_{\mathbf{x}}[t-1] + \\ &+ \rho(\mathbf{I_x}(\mathbf{x}) - \mu_{\mathbf{x}}[t])^T(\mathbf{I_x}(\mathbf{x}) - \mu_{\mathbf{x}}[t]) \end{aligned} \quad (2)$$

where ρ is the adaptation learning rate: $\rho \propto G_{\mathbf{x}}(\mathbf{I}(\mathbf{x}))$.

However, our approach differs from [9] in an important way. In [9], background classification is performed when the pixel value falls within 2.5 standard deviations of the mean of the Gaussian. Otherwise, it is classified as foreground. Our approach differs in that the foreground is not classified as an exception to the background model. Instead, we prefer to express the problem in a Bayesian form. To do so, first we need to model the foreground process.

3.2. Uniform Foreground Model

The foreground process can be modeled using histograms, Gaussians or any other *pdf*. However, we simply use a uniform *pdf* to model the foreground process in each pixel, which is the best we can do if we know nothing about the foreground process in the scene.

Since a pixel admits 256^3 colors in the RGB color space, we model its *pdf* as

$$U_{\mathbf{x}}(\mathbf{I}(\mathbf{x})) = \frac{1}{256^3} \quad (3)$$

3.3. 2D Fore/Background Classification

Once that the foreground and background likelihoods of a pixel have been introduced, and assuming that we have some knowledge of foreground and background prior probabilities, $P(\phi)$ and $P(\beta)$[1], respectively, we are now in position to further discuss how the 2D-classification process can be done.

The probability that a pixel $\mathbf{x}$ belongs to the foreground (ϕ), given an observation $\mathbf{I}(\mathbf{x})$, can be expressed in terms of the likelihoods of the foreground and background processes as follows

$$P(\phi|\mathbf{I}(\mathbf{x})) = \frac{P(\phi)p(\mathbf{I}(\mathbf{x})|\phi)}{p(\mathbf{I}(\mathbf{x}))}. \quad (4)$$

In order to compute (4), the unconditional joint probability density ($p(\mathbf{I}(\mathbf{x}))$) can be expressed in terms of the conditional distributions as

$$p(\mathbf{I}(\mathbf{x})) = P(\phi)p(\mathbf{I}(\mathbf{x})|\phi) + P(\beta)p(\mathbf{I}(\mathbf{x})|\beta). \quad (5)$$

Then, in the case of the models described in the previous section, (4) is

$$P(\phi|\mathbf{I}(\mathbf{x})) = \frac{P(\phi)\frac{1}{256^3}}{P(\phi)\frac{1}{256^3} + P(\beta)G_{\mathbf{x}}(\mathbf{I}(\mathbf{x}))}, \quad (6)$$

and $P(\beta|\mathbf{I}(\mathbf{x})) = 1 - P(\phi|\mathbf{I}(\mathbf{x}))$.

Thus, a pixel is classified into the foreground class using maximum a posteriori (MAP) if $P(\phi|\mathbf{I}(\mathbf{x})) > \frac{1}{2}$ is satisfied. Alternatively, the following test can also be used:

$$P(\phi)P(\mathbf{I}(\mathbf{x})|\phi) > P(\beta)P(\mathbf{I}(\mathbf{x})|\beta), \quad (7)$$

which is faster, since the denominator in (4) does not have to be computed. We can graphically express the problem in Fig. 2, assuming a grayscale color space.

Indeed, more elaborated foreground models would allow obtaining better results. Better foreground models can be obtained by using a tracker so that the models of each foreground entity can be correctly updated along the time. However, a tracker would incorporate undesirable complexity to the system entailing a CPU overload for the devices that we are targeting for the future. In any case,

[1] Foreground and background priors depend on the application. However, approximate values can be easily obtained for each application by manually segmenting the foreground in some images, and averaging the number of segmented points over the total.

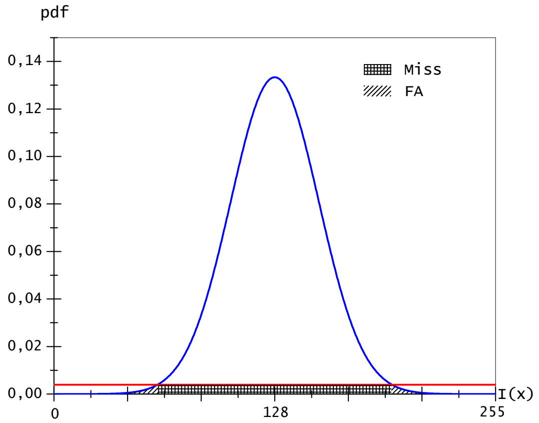
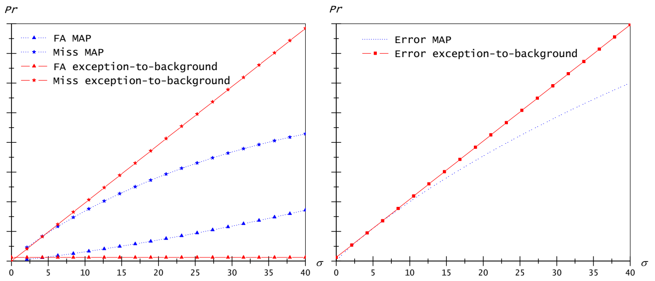


Fig. 2. Probability density functions of a 1D-Gaussian (in grayscale color space), and a uniform ($\frac{1}{256}$) distribution. The best decision possible, assuming equiprobable priors, is choosing background (β) for all those values where the Gaussian is over the uniform function. The figure also indicates the intervals where the two types of possible errors (false alarms -FA- and misses) happen. The integral of the likelihoods by these intervals give the system's probabilities of FA and miss.

a MAP setting outperforms exception-to-background methods even with this naive foreground characterization, as Fig. 3 shows[2]. Finally, the Gaussian model is adapted using (2), when the pixel is classified into the background.



(a) (b)

Fig. 3. The figures compare the error probabilities (false alarm -FA- and miss) of the MAP Bayesian setting and the exception-to-background setting for different values of σ. Error probabilities have been calculated by taking the integrals of foreground and background likelihood functions by the integration intervals where the errors exist (see Fig. 2). Altogether, after summing both probabilities in (b), the error rate in the exception-to-background setting is always larger than in the Bayesian setting.

3.4. Shadow Removal

Once the foreground objects pixels have been identified, a speckle noise removal filter is applied to suppress remaining isolated noisy foreground pixels. Then, an additional scheme [10] is applied to find out if some of these foreground pixels correspond to areas likely to be cast shadows or specular reflections. The working mechanism of this scheme is the following: As the first step, we evaluate the variability in both brightness and color distortion [11] between the foreground pixels and the adaptive background, and possible shadows and highlights are detected. It was observed though that this procedure is less effective in cases that the objects of interest have similar colors to those of presumed shadows. To correct this, an assertion process comparing the gradient / textures similarities of the foreground pixels and corresponding background is incorporated. These processing steps, effectively removing cast shadows, also invariably delete some object pixels and distort object shapes. Therefore, a morphology-based conditional region growing algorithm is employed to reconstruct the objects shapes. This approach gives favorable results compared to the current state of the art to suppress shadows / highlights.

4. 2D VIDEO + BINARY DEPTH CODING

Once the depth map is estimated, it has to be encoded into the main video stream and transmitted efficiently to the receiver. Since we are considering a videoconferencing application which is backwards compatible with the existing systems, the main video stream is encoded using the H.263++ codec.

The following step consists in encoding the depth map in a suitable way so as to preserve the compatibility without adding too much overhead. The solution considered in our approach consists in making use of the user data field defined in Annex W of H.263++ specification to enclose the related depth map as additional information in each frame.

The user data is defined by three main fields: a header, the size of the depth map in bytes, and the depth map coded with a run length encoder (RLE). The header in our case consists in a set of four bytes, 'T','S','V','C'. It will be used by any compatible decoder to detect the presence of a depth map and therefore to extract and process it accordingly. The size of the depth map is mandatory since the decoder must know the length of the 3D information enclosed in the user data to correctly extract it. Finally, the depth map, previously compressed in run length, will be copied into the user data. The use of an RLE is motivated by the fact that the depth map is a binary map. The compression offered by this algorithm is far more efficient than, for instance JPEG, though it is a lossless scheme. Besides, the image quality is preserved which avoids the use of post-processing filters at the receiver side.

This system ensures backward compatibility. If the decoder is not able to read the user data field or if it does not understand the header, the 3D information will be discarded and only the 2D video will be displayed. On the other hand, if the decoder is compatible with the new service, it will be able to extract the depth map accordingly, enabling the 3D videoconferencing system.

This solution presents various advantages, it is fully backwards compatible, relatively easy to implement and respects the existing standard. Also, the overhead can be controlled by the compression rate of the depth map. Besides, the images can be down-sampled with respect to the original view since it has been demonstrated that depth information does not require as much resolution as the 2D information to achieve visually satisfying results. In our case, we have about 15% video coding overhead using QCIF video and SQ-CIF depth, operating at 10 frames per second. The overall overhead for the whole service, including audio and signaling protocols, is about 10%.

[2] Probabilities of miss in the exception-to-background setting are computed assuming the same uniform foreground model used in the Bayesian setting.

This solution represents thus an elegant alternative while future *ad hoc* standards such as MPEG-C part 3 appear.

5. VIDEO SYNTHESIS AND DISPLAY

Regarding the video synthesis, a Philips 3D display is used. This solution consists in an auto-stereoscopic 42" display. No shutter glasses are needed to enjoy the 3D effect. Moreover, the Philips technology "WOWvx" implements a multi-view system that displays nine views in nine directions, ensuring a wide angle of vision.

The media player at the receiver side must however decode the incoming video stream in a format compatible with this display. Therefore, the main video frames are first decoded along with the related depth maps. Then, the two resulting images are merged to create the side-by-side picture accepted by the display. Finally, the display performs the synthesis of the different views in real-time, creating a 3D effect from the reference image and the depth map.

6. EXPERIMENTAL RESULTS AND DISCUSSION

The system has been extensively tested and shown in our lab at Barcelona (see Fig. 4), showing very promising results. The tests have been conducted using a commodity WebCam operating at 10 QCIF frames per second. CPU load is about 10% on an Intel Xenon CPU clocked at 3GHz and memory consumption is kept under 1MB since the adaptive foreground segmentation algorithm does not maintain a copy of old frames.

The Bayesian approach has proved to perform better than traditional exception-to-background methods in our experiments. However, there are still some foreground pixels which may not be detected in those regions whose colors are very similar to their counterparts in the background. The problem is hardly noticed at the display side, though, since the display internally applies a low-pass Gaussian filter to the depth map. Moreover, we have observed that cameras with higher quality CCDs hardly suffer from this problem, since they are better at discriminating colors.

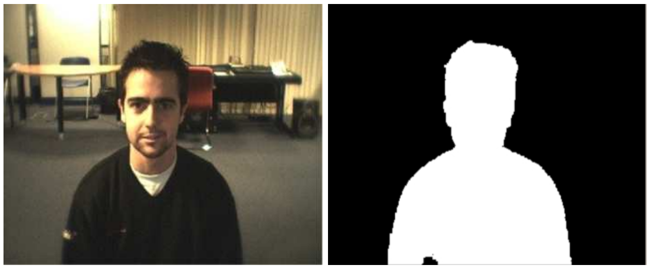

(a) (b)

Fig. 4. Figures (a) and (b) show an example of original image and computed depth map, respectively.

7. CONCLUSION AND FUTURE WORK

The main contributions of our system are the use of a conventional inexpensive camera at the sender side, low bandwidth overhead, network transparency and low CPU usage. The 3D videoconference service offers a very good 3D sensation that surprises users without previous 3D experience.

We believe that the presented scheme can be improved in several ways. One possibility is to add depth detail to the binary disparity mask we obtain in our algorithm, by means of fitting a general 3D human body model to the mask. Indeed, the use of body models will also help reducing the number of misses and false alarms. A second alternative is to embed our design in a special purpose DSP or FPGA architecture, which can be highly tailored to our algorithm's particular computational needs at a low cost. We are currently working in this later alternative. Also, we are planning to encode the user data following the ITU Recommendation T.35. This recommendation enables labeling data based on a country code and a terminal provider code.

8. REFERENCES

[1] O. Schreer, E.A. Hendriks, J.M. Schraagen, J. Stone, E. Trucco, and M. Jewell, "Virtual team user environments - a key application in telecommunication," *Proc. of eBusiness and eWork*, pp. 916–923, October 2002.

[2] H. H. Baker, N. Bhatti, D. Tanguay, I. Sobel, D. Gelb, M. E. Goss, W. B. Culbertson, and T. Malzbender, "Understanding performance in coliseum, an immersive videoconferencing system," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 1, no. 2, pp. 190–210, 2005.

[3] Wei-Wao Chen, , Herman Towles, Lars Nyland, Greg Welch, and Henry Fuchs, "Towards a compelling sensation of telepresence: Demonstrating a portal to a distant (static) office," in *Proceedings IEEE Visualization 2000*, T. Ertl, B. Hamann, and A. Varshney, Eds., 2000, pp. 327–333.

[4] H. Towles, W.-C. Chen, R. Yang, S.-U. K., H. F. N. Kelshikar, J. Mulligan, Daniilidis K, and et al., "3d tele-collaboration over internet2," 2002.

[5] Kiriakos N. Kutulakos and Steven M. Seitz, "A theory of shape by space carving," *Int. J. Comput. Vision*, vol. 38, no. 3, pp. 199–218, 2000.

[6] A. Laurentini, "The visual hull: A new tool for contour-based image understanding," *Proc. Seventh Scandinavian Comperence on Image Processing*, pp. 993–1002, 1991.

[7] J.L. Landabaso and M. Pardàs, "Foreground regions extraction and characterization towards real-time object tracking," in *Proceedings of Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI '05)*, 2005.

[8] Daniel Scharstein and Richard Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[9] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. on Pattern Anal. and Machine Intel.*, vol. 22, no. 8, pp. 747–757, 2000.

[10] J. L. Landabaso, M. Pardas, and L.-Q. Xu, "Shadow Removal with Blob-based Morphological Reconstruction for Error Correction," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, Philadelphia, PA, USA, March 2005.

[11] T. Horpraset, D. Harwood, and L. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," *Proceedings of International Conference on Computer Vision*, 1999.