

Research Article

Transmission of 3D Scenes over Lossy Channels

Pietro Zanuttigh, Andrea Zanella, Federico Maguolo, and Guido Maria Cortelazzo

Department of Information Engineering, University of Padova, 35131 Padova, Italy

Correspondence should be addressed to Pietro Zanuttigh, pietro.zanuttigh@dei.unipd.it

Received 26 May 2009; Accepted 23 September 2009

Academic Editor: Irene Cheng

Copyright © 2010 Pietro Zanuttigh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper introduces a novel error correction scheme for the transmission of three-dimensional scenes over unreliable networks. We propose a novel Unequal Error Protection scheme for the transmission of depth and texture information that distributes a prefixed amount of redundancy among the various elements of the scene description in order to maximize the quality of the rendered views. This target is achieved exploiting also a new model for the estimation of the impact on the rendered views of the various geometry and texture packets which takes into account their relevance in the coded bitstream and the viewpoint required by the user. Experimental results show how the proposed scheme effectively enhances the quality of the rendered images in a typical depth-image-based rendering scenario as packets are progressively decoded/recovered by the receiver.

1. Introduction

Free Viewpoint Video (FVV) and 3DTV are novel research fields that aim at extending the possibilities of traditional television, allowing the viewers to watch a dynamic three-dimensional scene from any viewpoint they wish instead of just the viewpoint chosen by the director. The development of such a new service type is still at early stage; nonetheless it is expected to become a reality in the next few years and then to rapidly gain in popularity.

The realization of a 3DTV streaming service basically requires four main operations: the acquisition of the 3D scene, the compression of the data, their transmission, and finally their visualization at client side. A common assumption is that the description of a three-dimensional (static or dynamic) scene is made by two key elements, the geometry description and the color (or texture) information.

The color information can be represented by means of a set of views (or video streams for dynamic scenes) of the scene corresponding to the cameras' viewpoints. These images (or videos) are, then, compressed and transmitted by adapting the highly-scalable techniques developed for standard images and videos [1], for example, H.264 or JPEG2000.

The geometry information may be coded in different ways. Three-dimensional meshes are a common representation for geometry and many recent works focus on how to transmit them in a progressive and robust way over band-limited lossy channels [2, 3]. An alternate common solution, specially used in free viewpoint video, is to represent texture by a set of images (or videos) of the scene and geometry by depth maps. Depth maps are greyscale images that associate to each pixel a range value, corresponding to the distance between the viewpoint and the camera. This allows to reproject the available images on new viewpoints, according to the so-called "Depth-Image-Based Rendering" (DIBR) approach. This approach makes it possible to reuse the same standard image (or video) compression and transmission techniques both for texture and geometry data [4, 5], thus greatly simplifying the service architecture.

Nonetheless, the transmission and interactive browsing of 3D scenes introduces new challenges compared to standard image and video streaming. A relevant issue is that the impact of the different elements of the geometry and texture description on the rendered views dynamically changes with the viewpoint. In this regards, an open research issue is how to split the connection bitrate between texture and geometry in order to maximize the quality of the service [6, 7].

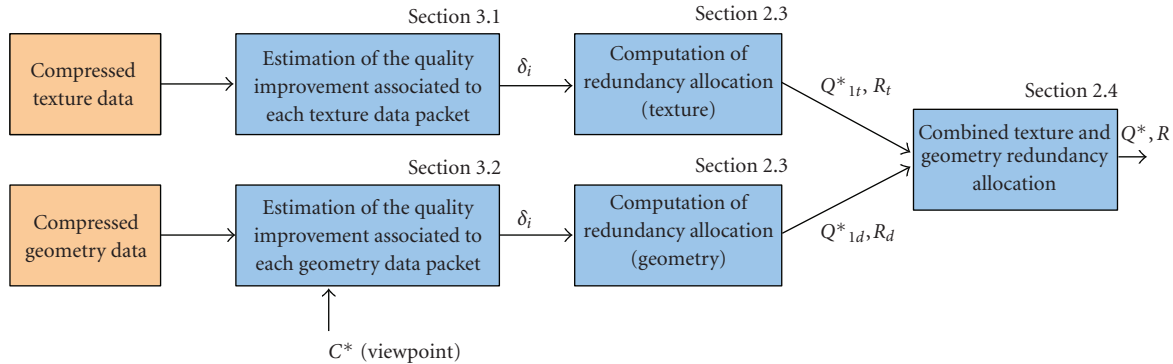


FIGURE 1: Block diagram of the proposed redundancy allocation scheme. The numbers over the blue boxes indicate the corresponding section in the paper.

The 3D streaming becomes even more challenging in presence of unreliable connections because packet losses may severely degrade the quality of the reconstructed content. Several methods have been proposed for the robust transmission of 3D models over lossy channels [3, 8–10] and of course many others exist for the transmission of image (texture) data [11]. However the combined transmission of both kinds of data over lossy channels is still a quite unexplored field. One of the few studies on the effects of packet losses in the combined transmission of texture and geometry data is presented in [12]. It is worth pointing out that previous literature mainly considers triangular meshes while in this paper geometry is represented as compressed depth maps. Although the transmission of 3D scenes can be performed upon either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) transport services, most of today implementations actually prefer the reliable transport service offered by TCP, which avoids the performance degradation due to packet losses. This choice is well suited to reliable and high-bitrate connections, where packet losses are rare so that the recovery and congestion control mechanisms of TCP do not impair the fluency and quality of the multimedia stream. However, in the perspective of offering 3D browsing and video services to multicast groups which reach a potentially wide population of users with heterogeneous connection capabilities, including unreliable and medium-to-low bitrate wireless connections, the reliable transport service offered by TCP will likely fail in providing the required trade off between perceived image quality and latency.

Another approach consists in abandoning the reliability of TCP in favor of a solution that employs an error recovery mechanism atop the best-effort transport service provided by UDP. A possible solution along this line consists in protecting the source data with an Unequal Error Protection (UEP) scheme, which basically consists in assigning redundancy to the different parts of the original bitstream in proportion to the importance of that part on the quality of the reconstructed view [13, 14]. UEP schemes are particularly suitable when the use of packet retransmission techniques is impractical, either because of delay constraints or in the case of multiple receivers, as for multicast transmissions. Clearly, the advantages in terms of packet loss resilience

provided by UEP schemes come at the cost of an increment in the communication overhead due to the transmission of redundancy packets. Therefore, the main problem when designing a UEP scheme is to cut the best tradeoff between error protection and overhead or, from another perspective, between quality of the delivered content and latency.

This paper focuses on the transmission stage of 3D scenes over lossy channels, such as the wireless ones. We assume that both texture and geometry data are compressed in a scalable way and transmitted over a lossy channel, using the UDP transport protocol.

We first propose a UEP scheme explicitly designed for multilayer source encodings. Such a scheme is then applied to the texture plus depth map encoding technique considered in this paper. In this way we determine the distribution of a prefixed amount of redundancy between texture and geometry packets that maximizes the quality of the rendered scene in presence of losses of texture and geometry information. Figure 1 shows a block diagram of the proposed approach (the number over each block indicates the paper section describing it). The computation of redundancy allocation requires two main steps: the estimation of the impact of each data packet on the rendered images and the optimal distribution of the redundancy packets based on this information. In this paper a new model to describe the distortion on the rendered views due to texture and geometry losses in function of the required viewpoint is introduced. Its accuracy in describing the impact of texture and geometry losses on the rendered scene was assessed by a number of experiments with varying locations of the losses in the coded bitstream and different viewpoints required by the user. Such a model is used to compute the quality improvement associated to each texture and geometry data packet. This information is then used in our UEP scheme in order to compute the amount of redundancy to assign to the various scene description elements. The effectiveness of the proposed UEP scheme was tested by using an experimental testbed where a server transmits views of a 3D scene to a remote client through a connection that drops packets in a controllable manner. In order to appreciate the trade off between scene quality and latency the performance of the UEP scheme is compared against that of

a simple protection scheme, which protects only the basic layers, and that of unprotected transmission. The results reveal that even a limited number of redundancy packets, as long as appropriately allocated to the different layers, can significantly improve the quality of the reconstructed scene at every time.

In summary, the main contributions of the manuscript are threefold: the design of an UEP scheme that jointly protects texture and depth information, the definition of a simple method to estimate the relative importance of depth and texture to be used for driving the UEP scheme, and, finally, the experimental evaluation of the UEP scheme in a realistic case study.

The paper is structured as follows. Section 2 presents a UEP scheme for the transmission of three-dimensional data over lossy connections. Section 3 presents a model to estimate the impact of texture and geometry packet losses on the rendered views. Section 4 presents the experimental results and Section 5 draws the conclusions.

2. Unequal Error Protection Scheme for 3D Streaming

In this section we propose a scheme to allocate a prefixed redundancy budget among the different layers representing the scene, in such a way that the service quality experienced by the end user is maximized. For the sake of generality, the Unequal Error Protection (UEP) scheme is designed by considering an abstract and rather general source model, which may apply to different multimedia sources and to 3D scene browsing, in particular.

2.1. Source Model. We suppose that the multimedia source, generically referred to as *scene* in the following, is encoded in a basic layer plus $L - 1$ enhancement layers. The basic layer carries the fundamental information from which it is possible to reconstruct a rough version of the scene. The enhancement layers, instead, contribute to progressively improve the quality of the content reconstructed at the receiver. Note that, in the case of 3D scenes transmission, this model separately applies to the texture and depth information; therefore there will be two sets of layers, one for color information and one for depth. In Section 2.4 it will be shown how to exploit this model for 3D scene browsing and in particular how to combine the depth and texture optimization procedures. Each enhancement layer is differentially encoded with respect to the previous one, so that an error in a quality layer recursively propagates to all the upper layers. For the sake of simplicity, we assume that the quality of the reconstructed scene improves only upon reception of a *complete* quality layer, whereas partial layer reception does not provide any quality enhancement. According to our experiments, this assumption is rather pessimistic since we have observed a limited quality improvement even for partially recovered layers. Therefore, the UEP scheme based on this simplified model will naturally tend to overprotect the layers with larger size, which are more likely affected by errors. Nonetheless, since the largest quality

improvement comes from the lower layers which get most of the redundancy independently of their size, the impact of this approximation on redundancy allocation is not very relevant.

Let us denote by q_i the function that describes the scene's quality after the correct decoding of the first i layers. Function q_i can be measured in terms of Peak Signal-To-Noise Ratio (PSNR), complementary Mean Squared Error (MSE), or any other metric monotonically increasing with the perceived quality. Function q_i is supposed to be known by the multimedia server for each scene transmitted to the client. The actual computation of the values of q_i for each element of the texture and depth information bitstreams is the subject of Section 3.

2.2. Transmission Model. We focus our attention on a unicast connection (or a single path of a multicast distribution tree). We assume that data are transmitted using the UDP transport protocol, in order to avoid the unpredictable delay that may derive from the congestion-control and loss-recovery mechanisms of TCP. Differential encoding generally yields to quality layers of different size. We assume that data are divided in packets of equal size L and we denote by k_i the number of packets of layer i . Therefore, the transmission of the *source* data for the whole scene all together requires

$$K = \sum_{i=1}^L k_i \quad (1)$$

packets. We assume that each packet can be lost with constant probability P_{loss} , independently of the other packets, according to the classical Bernoulli model. This type of error pattern may be observed, for instance, in case of a congested path where intermediate routers implement Random Early Dropping techniques [15, 16], or in presence of an unreliable wireless link. Every lost packet represents an *erasure*, that is, a missing packet in the stream. Erasures are easier to deal with than bit errors since the exact position of missing data is known, thanks to the sequence numbers that can be added by the transmission protocol as part of the UDP datagram payload. (When the transmission is realized by using Real Time Protocol (RTP) or the JPIP protocol, packets numbering is natively provided.) In order to increase resilience to erasures, the multimedia server adds to the k_i data packets of each layer i further r_i redundancy packets obtained by a systematic Forward Error Correction (FEC) code for erasure channels [17]. In this way, *any* subset of k_i packets suffices to reconstruct the source data. In other words, this code allows the receiver to recover from up to r_i packet losses in the group of $k_i + r_i$ encoded packets of layer i . When more than r_i packets are lost, no erasure can be recovered. (Note that, whereas the mathematical model here considered for redundancy allocation simply ignores these packets, the software tools used for the experimental results described in Section 4 do exploit even partially received layers.) Hence with independent packet losses, the probability of complete recover of layer i is equal to

$$p_i(r_i) = \sum_{\ell=0}^{r_i} \binom{k_i + r_i}{\ell} P_{\text{loss}}^{\ell} (1 - P_{\text{loss}})^{k_i + r_i - \ell}. \quad (2)$$

2.3. Redundancy Allocation Algorithm. In order to strike a balance between overhead and robustness to erasures, we set to R the total number of redundancy packets for each scene. The problem, then, is how to best distribute this redundancy among the L layers.

Here, the optimality criterion we consider consists in maximizing the average quality level of the scene reconstructed by the receiver. According to our source model, scene quality progressively increases with the reception of the different layers and stops when a layer is not completely recovered by the receiver. Let δ_i denote the quality increment associated to the i th layer, so that $\delta_1 = q_1$ and $\delta_i = q_i - q_{i-1}$ for $i = 2, 3, \dots, L$. Furthermore, let χ_i be the indicator function of event {Layer i is correctly decoded}, which is equal to one if the event holds true and zero otherwise. The quality of the scene reconstructed by the receiver can then be expressed as

$$\tilde{q}_1(\boldsymbol{\rho}_1) = \delta_1 \chi_1 + \delta_2 \chi_1 \chi_2 + \dots + \delta_L \chi_1 \chi_2 \dots \chi_L = \sum_{i=1}^L \delta_i \prod_{j=1}^i \chi_j, \quad (3)$$

where $\boldsymbol{\rho}_1 = [r_1, r_2, \dots, r_L]$ denotes the vector of redundancy packets allocated to layers 1 to L . Taking the expectation of (3) we then get the *mean* scene quality at the receiver after layers 1 to L have been processed, that is,

$$Q_1(\boldsymbol{\rho}_1) = \mathbb{E}[\tilde{q}_1(\boldsymbol{\rho}_1)] = \sum_{i=1}^L \delta_i \prod_{j=1}^i \mathbb{E}[\chi_j] = \sum_{i=1}^L \delta_i \prod_{j=1}^i p_j(r_j), \quad (4)$$

where the result follows from the independence of the indicator functions χ_i and the fact that $\mathbb{E}[\chi_j] = \Pr(\chi_j) = p_j(r_j)$. The objective of the allocation algorithm is to maximize (4) by optimally allocating the R redundancy packets. In other words, we wish to attain

$$Q_1^*(R) = \max_{|\boldsymbol{\rho}_1|=R} Q_1(\boldsymbol{\rho}_1) = \max_{|\boldsymbol{\rho}_1|=R} \left[\sum_{i=1}^L \delta_i \prod_{j=1}^i p_j(r_j) \right]. \quad (5)$$

where $|\boldsymbol{\rho}_1| = \sum_{i=1}^L r_i$. The vector $\boldsymbol{\rho}_1^* = [r_1^*, r_2^*, \dots, r_L^*]$ that attains $Q_1^*(R)$ is named the *optimal allocation policy* of the R redundancy packets. This optimization problem can be solved by dynamic programming [18]. To this end, we need to express the optimization problem (5) in a recursive form which makes it possible to obtain the general solution as combination of optimal subproblems solutions. Thus, we first rewrite (4) as

$$Q_1(\boldsymbol{\rho}_1) = p_1(r_1) \left(\delta_1 + \sum_{i=2}^L \delta_i \prod_{j=2}^i p_j(r_j) \right). \quad (6)$$

Then, by the functions $Q_h(\boldsymbol{\rho}_h)$ defined as

$$Q_h(\boldsymbol{\rho}_h) = \sum_{i=h}^L \delta_i \prod_{j=2}^i p_j(r_j), \quad \text{with } \boldsymbol{\rho}_h = [r_h, r_{h+1}, \dots, r_L], \quad (7)$$

we can rewrite (6) in the following recursive form:

$$\begin{aligned} Q_h(\boldsymbol{\rho}_h) &= p_h(r_h) (\delta_h + Q_{h+1}(\boldsymbol{\rho}_{h+1})), \quad h = 1, \dots, L-1, \\ Q_L(\boldsymbol{\rho}_L) &= p_L(r_L) \delta_L, \end{aligned} \quad (8)$$

where each function $Q_h(\boldsymbol{\rho}_h)$ can be interpreted as the average quality increment due to the decoding of layers from h to L when the redundancy packets are allocated according to vector $\boldsymbol{\rho}_h$. The optimization problem (5) can then be formulated in the following recursive manner:

$$Q_h^*(R_h) = \max_{0 \leq r_h \leq R_h} [p_h(r_h) (\delta_h + Q_{h+1}^*(R_h - r_h))] \quad (9)$$

for $h = 1, 2, \dots, L-1$, whereas

$$Q_L^*(R_L) = \max_{0 \leq r_L \leq R_L} [p_L(r_L) \delta_L]. \quad (10)$$

Equations (9) and (10) express optimization problem (5) as a Bellman equation that can be numerically solved by backwards induction [18]. In practice, the algorithm starts by evaluating the last term of the recursion, (10), for each possible value of $R_L \in \{0, 1, 2, \dots, R\}$. Then, it evaluates (9) for $h = L-1$ and for any value of $R_h \in \{0, 1, 2, \dots, R\}$. The algorithm iterates backwards till it determines the best allocation of the redundancy packets for each recursion step $h = 1, 2, \dots, L$. The policy that attains the optimal quality $Q_1^*(R)$ is the final allocation vector. A pseudocode that describes the operations performed by the algorithm is shown in Algorithm 1.

2.4. Adapting the UEP Scheme to 3D Scene Browsing. As mentioned above, the transmission of 3D scenes actually involves two types of data flows, namely, texture and depth map. To determine the best redundancy allocation for both texture and depth map packets we firstly apply the above described method to each stream, separately, and then we merge the results within a further optimization loop. More specifically, let $Q_{1D}^*(R_D)$ denote the quality corresponding to the optimal protection strategy for depth information as a function of the amount of redundancy R_D allocated to it. Let $Q_{1T}^*(R_T)$ and R_T represent the same quantities for texture data. The optimal quality $Q_1^*(R)$ achievable with R redundancy packets is then given by

$$Q_1^*(R) = \max_{R_D + R_T = R} Q_{1D}^*(R_D) + Q_{1T}^*(R_T). \quad (11)$$

The optimal redundancy distribution between the two elements of the scene description is given by the couple (R_D, R_T) that maximizes $Q_1^*(R)$ subject to the constraint on the total amount of redundancy packets $R_D + R_T = R$. Considering that the number of redundancy packets is limited and the algorithm of Section 2.3 is very fast, the solution can be easily found by performing an outer loop on all the (R_D, R_T) couples for which $R_D + R_T = R$. The two vectors $\boldsymbol{\rho}_1^*$ corresponding to R_D and R_T represent then the optimal allocation policy for both data streams. Note


```

/*
Redundancy allocation algorithm
Input:
L = number of layers
D(j) = quality increment of layer j
K(j) = number of packets of layer j
Ploss = Packet loss probability
Rtot = total number of redundancy packets
Output:
Ropt(j) = redundancy packets assigned layer
Q(j,h) = overall quality provided by layers j to L when
protected by a total redundancy of h
*/
function redund (L,D,K,Ploss,Rtot)
/* Variable initialization
r(j,h) = redundancy assigned to layer j given that
layers j to L have a total budget of h redundancy
blocks */
r = 0; Q=0;
for h=0 : Rtot,
    r(L,h) = h;
    Q(L,h) = binocdf(K(L)+h,h,Ploss)*D(L);
endfor
/* Recursion */
for j=L-1:-1:1,
    for h=0:Rtot,
        for g=0:h;
            p_jh=binocdf(K(j)+g,g,Ploss);
            Qtmp = p_jh*(D(j)+Q(j+1,h-g));
            if Qtmp > Q(j,h)
                Q(j,h) =Qtmp;
                r(j,h) = g;
            endif
        endfor
    endfor
endfor
/* Search for the overall best allocation of the total
redundancy Rtot */
Rres = Rtot;
for j=1:L,
    Qopt = 0;
    for h=0:Rres,
        if Q(j,h)>Qopt,
            Qopt = Q(j,h)
            Ropt(j) = r(j,h);
        endif
    endfor
    Rres = Rres-Ropt(j);
endfor
return Q
return Ropt
/* Binomial CDF */
function binocdf(n,m,p)
f= n!/(m!(n-m)!)*p^m*(1-p)^(n-m)
return f

```

ALGORITHM 1: Redundancy allocation algorithm.

of the values due to the two contributions. Nevertheless, experimental evidence in Section 4 (compare, e.g., Figures 19 and 20) shows how the additive model gives an estimate of the total distortion not too far from the actual values, at least for the considered cases. Therefore the approximation of (11) is sufficiently accurate, at least for the purposes of this paper (i.e., redundancy allocation). Further research will be devoted to the development of a more accurate way of combining the two quality values. In this connection it is worth noting that since the considered quality model provides just a rough approximation of the actual impact of texture and depth losses, the redundancy allocation provided by the proposed model is no longer optimal. Nonetheless, we experimentally observed that the redundancy allocation of the proposed UEP scheme is rather robust to variations of the quality measures. Therefore, in spite of its suboptimality, the proposed quality model is remarkably effective for redundancy allocation.

3. A Model for the Relevance of Depth and Texture Packets in 3D Streaming

The UEP scheme described in Section 2 requires the server to be able to determine the effects of the packet losses affecting different parts of the coded stream. In this section, we present a model to estimate the impact of the loss of each element of the compressed geometry and texture bitstream on the quality of the rendered images (i.e., how to estimate the q_i values in the model of Section 2).

Since both texture and depth information are represented as images or videos, before analyzing the three-dimensional case, it is useful to briefly recall a couple of basic characteristics of scalable compression, found in many current schemes.

- (i) The first packets to be transmitted typically correspond to the basic quality layers or to lower resolutions in multiresolution schemes. They usually have a much greater impact on visual quality than the subsequent ones. However an accurate model is rather difficult to obtain since it depends on the data that is being transmitted and on the selected compression standard (for JPEG2000 image compression an estimation strategy is presented in [20]).
- (ii) In some compression standards, like JPEG2000, the image can be decoded from any subset of the codestream. However, typically the loss of a packet can affect the usefulness of the following ones. In the video transmission case it is also necessary to consider that losses in the intra frames affect also all the subsequent frames predicted from them.

3.1. Loss of Texture Information. As far as the loss of texture information is concerned, the only difference with standard image transmission is that in our case the images are reprojected to novel viewpoints and this process can in principle change the impact of the lost packets.

that, as reported by perceptual studies, for example [19], the quality of the rendered scene is not always the sum

For example's sake we will illustrate this point referring to the case of JPEG2000; however similar results can be derived for other scalable compression schemes. In JPEG2000 images are decomposed by the wavelet transform into a set of subbands, and then the subband samples are quantized and compressed. The compressed data are stored in a set of code-blocks corresponding to the different subbands, spatial positions, and quality levels. The effects of a 3D warping operation on the distortion in the wavelet subbands have been analyzed in a previous work [5]. It was shown that the quantization error of a wavelet subband sample in the source view is mapped through wavelet synthesis, warping and further wavelet decomposition of the warped view to different samples and subbands of the decomposed warped image (in spatial locations close to the one of the reprojected quantized sample). The amount of distortion in the wavelet samples of the warped image can be computed exploiting a collection of precomputed weights. Without further detail, for the current discussion it suffices to note that the weight values depend both on the 3D warping operation and on the source and destination subbands of the code-block in the wavelet decomposition. The procedure to compute these weights is described in [5]. The distortion on each sample of the warped image can so be approximated by multiplying the distortion in each sample of the transmitted image (prior to warping) by the corresponding weighting coefficients and by summing all the contributions falling on that sample. The total distortion in the rendered sample at location \mathbf{p} can thus be approximated by

$$D_{\text{quant},d}^{i \rightarrow *}[p] = \sum_b W_{b \rightarrow d}[p] \cdot D_b^i \left[\left(\mathcal{W}_{b \rightarrow d}^i \right)^{-1}(p) \right], \quad (12)$$

where $D_b^i[k]$ denotes the mean squared distortion at location \mathbf{k} in subband b of the source view V^i and the weights $W_{b \rightarrow d}[p]$ represent how the distortion is mapped from the source subband b in V^i to the target subband d on the rendered view V^* on the basis of the surface warping operator \mathcal{W}^i . We are using $(\mathcal{W}_{b \rightarrow d}^i)^{-1}$ to indicate the operator which maps locations \mathbf{p} in the warped resolution subband d back to the corresponding location in subband b of V^i . A complete discussion of this framework can be found in [5] and is behind the scope of this paper. What is important to observe here is that the warping operation could change the impact of a lost packet, expanding the relevance of some packets and reducing the one of others. This effect is due to the fact that the image regions corresponding to some packets may shrink while other may enlarge. However the changes in the distortion measured on the rendered views due to such shrinkages or expansions are relevant only for large viewpoint shifts. Simulation results clearly show that these effects do not have such a big impact when averaged on the whole image. In typical 3D video setups where the cameras are quite close each other, the distortion on the rendered view due to a random packet loss on texture data can indeed be considered almost independent of the selected viewpoint. Future research however will also analyze the impact of warping in selecting the amount of redundancy to be assigned to the various elements.

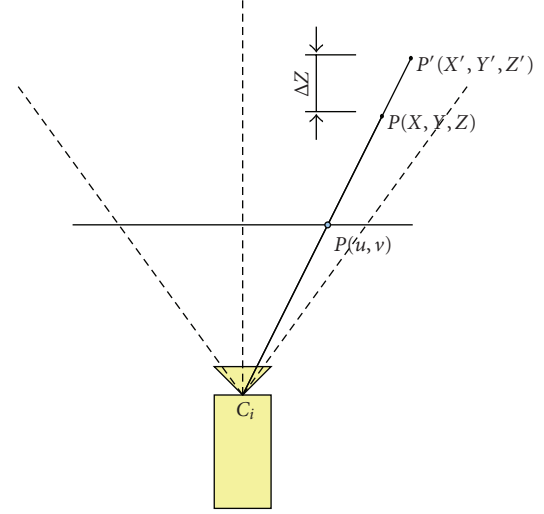


FIGURE 2: Pinhole camera model with depth information.

3.2. Loss of Depth Information. In 3D video systems depth information is used in order to allow the warping of the video stream corresponding to a camera to a novel viewpoint corresponding to a “virtual” camera that is not within the set of the available viewpoints. This is what allows the user at client side to observe the 3D scene from any viewpoint. One of the main difficulties in compressing and transmitting 3D data is to understand how uncertainty in the depth information, due to compression or network issues, affects the reconstruction of arbitrary views from novel viewpoints. Depth maps can be compressed as standard grayscale images. From image (or video) compression results it is possible to understand how packet losses or lossy compression affect the pixel values in such images (for the case of JPEG2000 an efficient estimation strategy is presented in [20]). However such a distortion should first be mapped to the representation of the scene geometry and from the scene geometry to the distortion on the rendering of novel viewpoints.

Let us denote with Cam^i an available camera with optical center in C^i and with Cam^* the target “virtual” camera with optical center in C^* . Figure 2 shows that each pixel $\mathbf{p} = (u, v)$ in the depth map of camera Cam^i corresponds to a point $\mathbf{P} = (X, Y, Z)$ in the 3D space. The depth map value represents the distance of \mathbf{P} from the optical center C^i projected on the camera's viewing direction. An error on the depth map pixel value will cause the sample to be mapped to a different point \mathbf{P}' along the optical ray. The error will so correspond to a displacement $\Delta\mathbf{P} = (\mathbf{P} - \mathbf{P}')$ of the 3D point position along the direction of the optical ray $C^i\mathbf{P}$. The 3D point is then mapped to pixel \mathbf{p}^* in the image space of camera Cam^* and the uncertainty in the position of the 3D point translates into a translational uncertainty $\Delta\mathbf{p}^*$ on the position of the sample within the warped image. The variance (error power) on the sample values of the depth map can be mapped to a

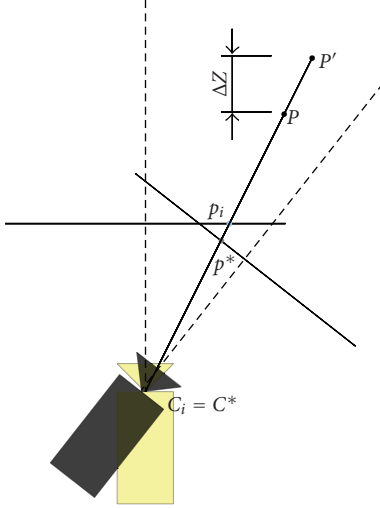


FIGURE 3: Rotation of the camera.

corresponding variance (error power) in the sample position by the following equation:

$$|\Delta \mathbf{p}^*|^2 = |\Delta Z[\mathbf{p}]|^2 g^{i-*}[\mathbf{p}], \quad (13)$$

where the factor $g^{i-*}[\mathbf{p}]$ depends upon the parameters of cameras Cam^i and Cam^* and on the scene geometry. A complete description of this model together with the equations for the computation of $g^{i-*}[\mathbf{p}]$ can be found in [5].

Unfortunately this model requires complex computations. If an accurate estimate of the distortion is not required, as it is the case of this work where distortion is used only to compute the relative amount of redundancy to be assigned to depth and texture, it is possible to approximate $g^{i-*}[\mathbf{p}]$ with simpler functions. To this end a few observations are in order.

- (i) In a pure rotational camera movement (see Figure 3), depth information is not relevant and displacement $\Delta \mathbf{P}$ does not affect the reconstruction of a novel view (indeed this is the only case where the novel view can be estimated without knowledge on depth information).
- (ii) A well-known result from stereo vision is that in a binocular system the projection of the same point in 3D space to two different views corresponding to a pair of cameras is shifted by an amount proportional to the distance between the two cameras (*baseline*). In particular in the simple configuration of Figure 4 it can be easily shown that the samples' shift from one view to the other is [21]

$$\mathbf{p}^* - \mathbf{p}_i = \frac{(\mathbf{C}^* - \mathbf{C}^i)f}{Z} = \frac{bf}{Z}, \quad (14)$$

where $b = (\mathbf{C}^* - \mathbf{C}^i)$ is the stereo system baseline and f is the camera's focal length. If we introduce an error $\Delta Z = Z' - Z$

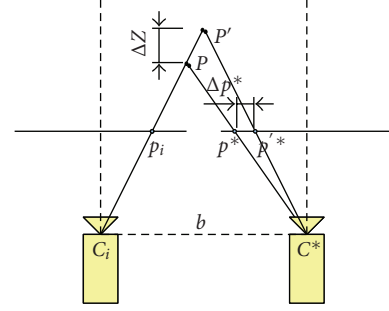


FIGURE 4: Translation of the camera.

on the depth values, the difference between the position of a point in the warped image computed with the correct depth value Z and the position of the same point in the warping computed with the corrupted depth value Z' is

$$\Delta \mathbf{p}^* = bf \frac{1}{Z} - bf \frac{1}{Z'} = bf \frac{\Delta Z}{ZZ'}. \quad (15)$$

This equation shows that in this case g^{i-*} is roughly proportional to the distance between the two camera viewpoints. Strictly speaking this holds only for cameras looking in the same direction. Furthermore term $1/(ZZ')$ leads to different shifts for objects at different distances from the viewpoint. However for small rotation angles and small camera movements in the direction of the optical ray, (15) can be assumed as a reasonable estimate of the displacement term. If a limited precision suffices, one may just average $g^{i-*}[\mathbf{p}]$ on the whole image and assume that it varies linearly with the baseline.

As expected real configurations are much more complex and include both camera translations and rotations. Experimental results, nevertheless, indicate that assuming that the rendering distortion depends only on the depth distortion and on the distance between the two camera viewpoints is reasonable for small viewpoint changes.

The final step is the conversion of the positional uncertainty to amplitude distortion of the samples in the rendered views. This operation can be performed using the method developed in [22]:

$$D^{i-*}[\mathbf{p}] = |\delta Z^*|^2[\mathbf{p}] \cdot g^{i-*}[\mathbf{p}] \cdot |\omega|^2 \cdot E^{i-*}[\mathbf{p}], \quad (16)$$

where $E^{i-*}[\mathbf{p}]$ is a local measure of the corresponding color image variance in the vicinity of \mathbf{p} and ω can be approximated by a constant factor. For the value and meaning of ω see [5, 22] where the proposed scheme is also applied independently on each of the subbands of the wavelet transform. It is worth noting that even if the cited works refer to JPEG2000, the distortion model of (16) is general and does not depend on the selected compression scheme. The distortion can then be summed up on all the image samples to get the estimation of the total distortion on the warped image corresponding to \mathbf{C}^* .

A final possibility is to simply warp the available view and a corrupted version of the depth information to build a novel view corresponding to a camera with the same orientation

placed at a distance $\Delta C = C^i - C^*$ from the original camera. The rendered view can then be compared with the image obtained by performing the same operation with the correct depth map. The procedure can be repeated for a limited set of viewpoints and the mean square error corresponding to all the others can be approximated on the basis of the camera's distances using the model of (15).

These approaches are very simple and provide a limited accuracy on the estimation of the distortion due to depth uncertainty; however they make it possible to build a practical real-time transmission system and provide a reasonable estimate of the relative weight of depth and texture data that can be used to select the amount of redundancy to be applied to each of them.

4. Experimental Results

In this section we describe the simulation environment used to test the performance of the proposed error correction scheme and we present the experimental results obtained by using both synthetic and real world multiview data with depth information. This section is organized in the following way: firstly we present the simulation environment, and then we analyze the effects of the loss of texture and depth packets. Finally, we show the performance of the proposed protection scheme.

4.1. Remote 3D Browsing with JPEG2000 and JPIP. As previously said many different transmission schemes for remote 3D browsing are possible. In this section for clarity's sake we briefly overview the client-server scheme of [5] which was used in order to test the performance of the algorithms presented in this paper. Let us emphasize that in spite the results of this paper were obtained with the system of [5], they are valid for every remote visualization scheme based on progressive transmission and on a depth-image based rendering scheme.

In the proposed approach the 3D scene description is available at server side as a set of images (or videos) and depth maps together with the corresponding acquisition points and camera parameters. To achieve an efficient browsing over band-limited channels, all the available information at server side, that is, both images and depth maps is compressed in a scalable way using JPEG2000. The server is able to select and transmit only the parts of the scalably compressed bitstreams that best fit the user's required viewpoint and the available bandwidth exploiting the rate-distortion optimization framework presented in [23]. The client then exploits the data received from the server to render the views required by the user during the interactive browsing. In [5] this is achieved by reprojecting all the available images onto the required viewpoint (that is usually different from the available ones) using depth information and then combining all the warped views into the requested rendering. This is achieved by using a novel multiresolution scheme based on the wavelet transform and on the estimation of the distortion on the rendered views coming from the different contributions.

The adopted transmission system relies on the JPIP interactive protocol [24] originally developed for the interactive transmission of JPEG2000 images over the internet. This protocol allows the server to decide what information needs to be transmitted to the client on the basis of the received requests. Figure 5 shows the architecture of the proposed transmission scheme and provides a framework for understanding the interaction between a JPIP server and its client. In JPIP image transmission the client just makes a request with the parameters of the view of interest (the region and resolution for an image can be replaced by the 3D viewpoint in our case) letting the server to decide the data most suited to satisfy the client's needs. The server streams JPIP messages to the client, where each JPIP message consists of a single byte-range from a single element (*data-bin*) of one of the compressed images or depth maps. JPEG2000 content can be rendered from any arbitrary subset of the precinct data-bins which might be available. This characteristic makes possible the rendering at client side completely asynchronous with respect to server communications and also makes the system robust with respect to packet losses and network issues.

4.2. Analysis of Packet Loss Issues. For the first test we used a synthetic scene of a cartoon character (*Goku*). In this case images and depth maps were generated from a synthetic 3D model. This makes available a ground truth and avoids the data uncertainties typical of real acquisition systems. The images were compressed in JPEG2000 at quite good quality (0.8 bit per pixel) and transmitted to the client using JPIP. For the *Goku* model we transmitted a single view with the corresponding depth map and analyzed the rendered views at 3 different viewpoints along a circle surrounding the object, respectively 5, 10, and 45 degrees apart from the original one obtaining the images of Figure 6. The camera positions are shown in Figure 7; note that the camera movement is both rotational and translational.

To test the performance of the proposed transmission scheme in a real environment we used the *Breakdance* sequence from Microsoft Research [25], commonly used to validate the performance of 3D video compression algorithms. This sequence is composed of 8 video streams acquired by a set of cameras placed along a nearly linear path (Figures 9 and 10). The dataset includes also depth information for each frame of each camera computed by stereo vision algorithms. As previously stated, all the transmitted information is compressed in JPEG2000 in a scalable way. We used 7 quality layers of increasing size. The impact of the packet losses was measured with the following procedure: the central view (Cam 4 in Figure 9) was transmitted together with the corresponding depth map and these data were used to render novel views. In particular the plots of this section show the results obtained by looking at the scene from the viewpoints of camera 3 (that is quite close to the available view) and of camera 1 (that is farther from the available viewpoint) in order to show also the dependency between the viewpoint's distances and the relevance of depth information presented in Section 3.2. To

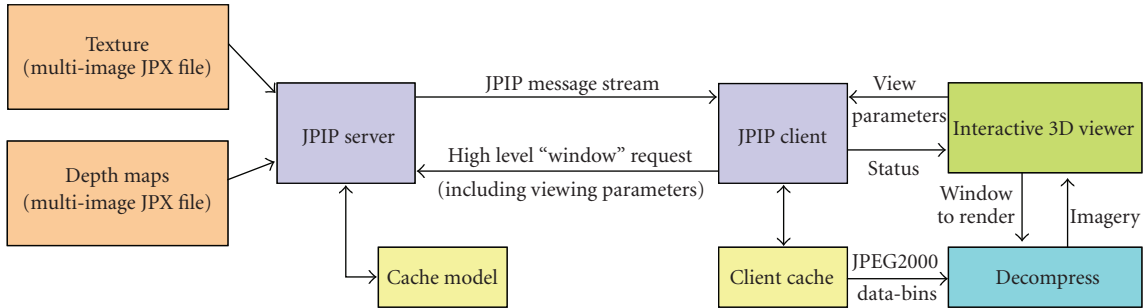


FIGURE 5: Architecture of the transmission system.

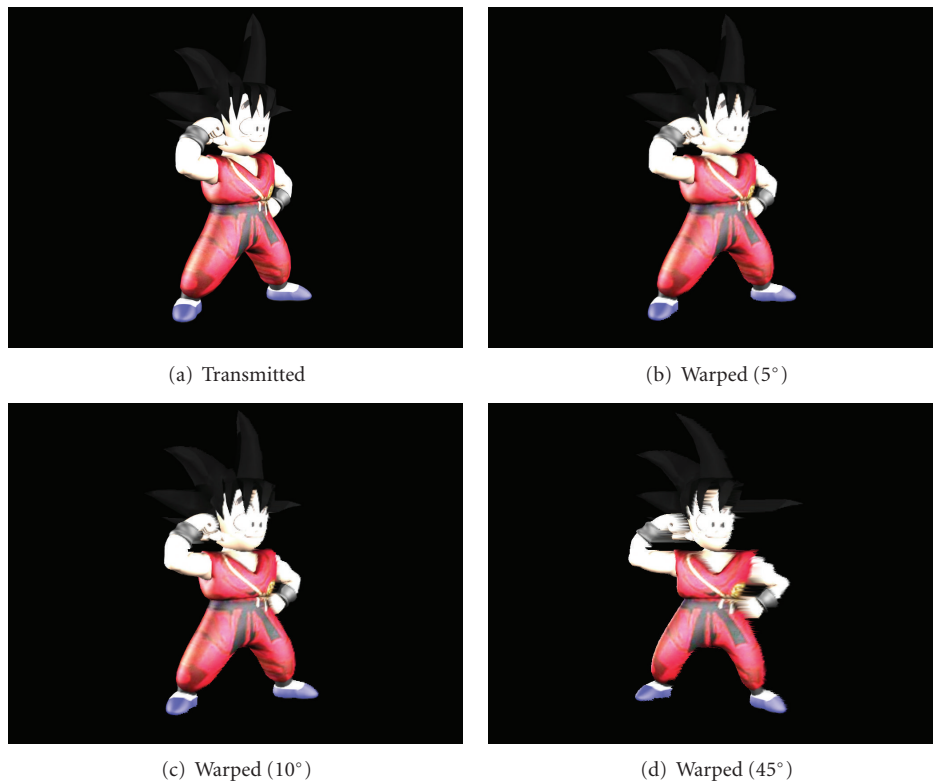


FIGURE 6: *Goku* model: transmitted image and rendered views obtained by warping the available one to novel viewpoints.

compute the rendered views' quality we took as reference the image obtained by transmitting the available data at maximum quality without any packet loss and performing the warping on these data. Even if it was possible to compare the images with the actual data from a camera in the novel viewpoint, this approach has the advantage of decoupling the loss due to transmission issues (that are the focus of this paper) from the distortion due to the warping algorithm, 3D reconstruction errors, occlusions, regions outside the reference camera field of view, and others. Examples of these issues are the black region on the side and the occluded regions close to the dancer's edge in Figure 14(b).

Figure 10 shows the PSNR corresponding to the loss of each single packet of texture information as a function of the lost packet position for the *Breakdance* data. Figure 11

instead refers to the *Goku* model and shows the distortion due to the loss of a batch of 5 consecutive texture packets. The plot shows clearly how the impact of packet losses is progressively less critical when losses occur towards the tail of the codestream. As previously said JPEG2000 divides the image in different codeblocks corresponding to different resolution levels, quality layers, and spatial regions. The JPIP server then estimates the distortion gain associated to each packet and transmits them in order of relevance. Therefore the first packets have a much greater importance than the last ones. A loss in the first layers specially in the lowest subbands data can lead to completely distorted images (Figure 12), while loss of higher resolution data can lead to blurred images. Moving towards the last quality layers the impact on the rendered views becomes less noticeable (Figure 13).

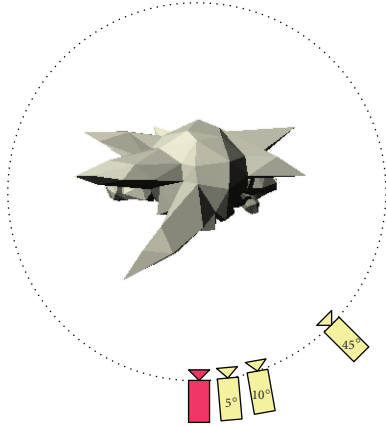


FIGURE 7: *Goku* model: camera positions (the available view and depth map correspond to the red camera).



(a)



(b)

FIGURE 8: *Breakdance* sequence: (a) sample frame of the sequence and (b) sample depth map.

In particular the loss of the first packet of the datastream (containing the main header and the key information for the first layer) results in the impossibility of decoding the whole image. This suggests that this packet must be protected with particular care. Another interesting observation is that the measured distortion is almost independent from the selected viewpoint (the plots of the warpings to viewpoints 1 and 3 are almost superimposed). Although the warping operation can affect the distortion due to the loss of each packet as shown in Section 3.2, on the average (specially if the

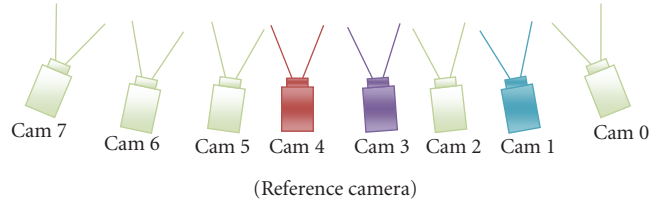


FIGURE 9: Camera positions for the *Breakdance* sequence (Courtesy of Microsoft Research). The colored cameras correspond to the viewpoints used for the experiments in this paper.

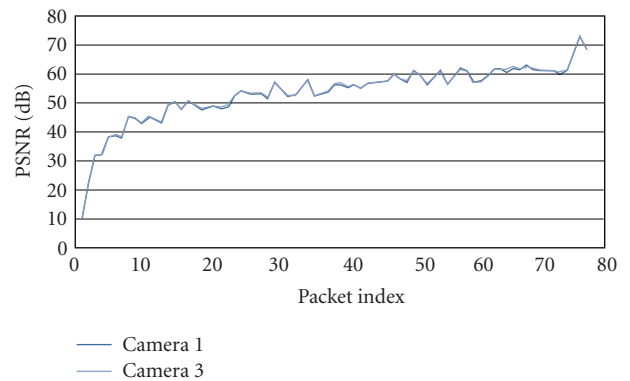


FIGURE 10: *Breakdance*: loss of a single packet of texture information (note how the two curves in the plot are almost superimposed).

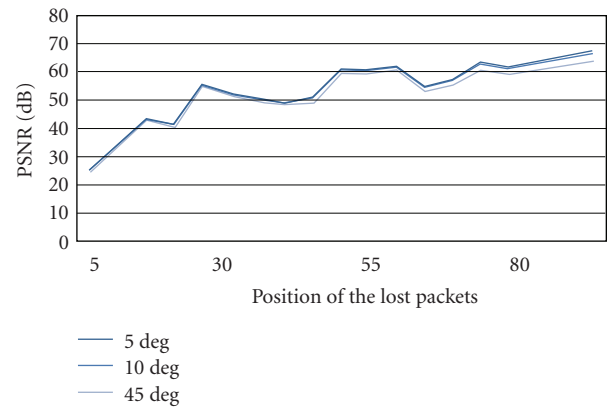


FIGURE 11: *Goku*: loss of a batch of 5 consecutive packets of texture information. The different curves correspond to the different viewpoints as shown in Figure 7. Note how the 5° curve is almost invisible because the 10° one is almost perfectly superimposed to it.



FIGURE 12: View of the *Goku* model corrupted by a loss in the first part of the texture codestream.

viewpoint is not very far from the available ones) distortion due to texture packet losses for practical purposes can be considered independent from the selected viewpoint. Figures 14(c) and 14(d) show an example of how texture losses artefacts remain quite similar after the warping (at least for small viewpoint changes).

Figures 15 and 16 concern instead the loss of depth packets. In particular, Figure 15 refers to the *Breakdance* data and shows the distortion due to a single depth packet loss. Figure 16 shows instead the distortion due to the loss of 5 packets of the *Goku* dataset. As shown in Section 3.2, the loss of depth packets causes samples to be misplaced in the 3D space and so to be warped in wrong positions in the rendered views. This of course impacts the quality of the rendered images; Figures 17 and 18 show a couple of examples of the artefacts due to this effect. Usually the edges of the objects (corresponding to the regions with depth discontinuities) experience the major impact from depth transmission errors. Note that also in this case the plots show the distortion in the rendered images, not the displacement of the samples or other measures in the 3D space. This allows to directly compare these results with the texture ones and it is also consistent with the target of the proposed scheme that is the maximization of the rendered views' quality and not the accuracy of the three-dimensional description.

There are two key differences with respect to the previous case.

- (i) The distortion still decreases with the packet index but the shape of the curve is less regular. This is due to the fact that JPIP streams the packets sorted by their impact on image distortion (therefore it considers depth maps just as regular images ignoring their 3D meaning and the way they will be used in the warping). Artefacts on the texture directly map to similar ones in the warped views, while the mapping of depth distortion to the warped views is more complex as pointed out in Section 3.2.

- (ii) In the depth case the measured distortion depends on the viewpoint: when warping to farther locations the same loss on depth data leads to larger errors in the warpings as shown in Section 3.2. Figures 18(c) and 18(d) clearly show how the same loss on the depth data leads to quite different results when warping to viewpoint 3 (close) or viewpoint 1 (farther). Note also how in the *Goku* setup, where the viewpoint change is much larger, the relative relevance of depth information is rather high if compared with the *Breakdance* sequence.

4.3. 3D Transmission over Lossy Channels with the Proposed FEC Protection Scheme. In this section we compare the image quality corresponding to the transmission of the image and depth data with and without error protection schemes. As shown in Section 4.2 the loss of the first block prevents image decoding and has a dramatic impact on average distortion, so we decided to compare three strategies:

- (i) Transmission without any protection scheme.
- (ii) A simple protection scheme that protects only the first packet in order to ensure that the image is decodable.
- (iii) The ad hoc protection scheme of Section 2.

To analyze the performance of the proposed transmission scheme we transmitted the information on the 4th view of the *breakdance* sequence over a network with random packet loss probability of 1%, 5%, and 10%. The received data are then rendered from the viewpoint of view 3 (quite close) and of view 1 (farther).

Figure 19 shows the Mean Squared Error (MSE) for a 10% random packet loss probability. The first two columns refer to the transmission of depth information (assuming that texture has been correctly received). In the case of transmission without any protection, performance is very poor because when the first packet is lost, the entire depth map becomes useless. In this case (assuming that the warping procedure cannot be performed without depth data), the MSE is equal to 650, which is an order of magnitude larger than the average MSE values obtained when the first packets are successfully decoded. Therefore, to preserve the graphs readability, we do not report in the plots the MSE for completely unprotected transmissions. Instead, our benchmark for assessing the quality gain due to smart redundancy allocation is provided by a very simple forward error correction scheme, called "PROT1," which protects only the first packet, guaranteeing its recovery probability very close to one. The performance obtained by PROT1 is shown by the right-most column of each group of bars in Figure 19. The middle column of each group ("FEC4") shows the performance of the proposed scheme with $R = 4$ redundancy packets, corresponding to a redundancy of 6% (there are 62 data packets for depth information in this frame). With just 4 packets of redundancy, most of them (three in this case) are allocated to the protection of the first packet. However the proposed protection scheme still

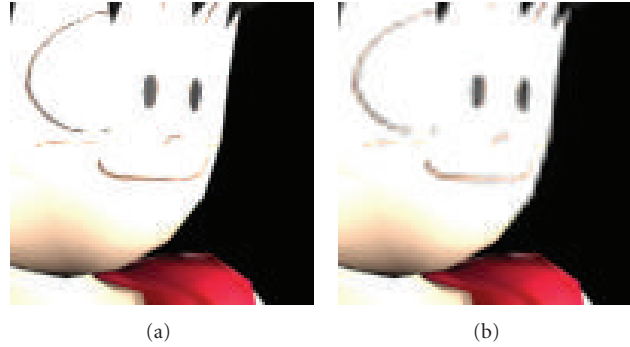


FIGURE 13: *Goku*: (a) detail of the character's face and (b) same detail corrupted by the loss of the 23th packet.

provides a small gain compared to the “PROT1” approach. Such redundancy is quite limited for a 10% packet loss and better results can be obtained using $R = 8$ packets of redundancy, corresponding to a 12% of data redundancy, and shown by the left-most column denoted as “FEC8.” In this way it is possible to protect a larger part of the depth codestream and the gain on the distortion in the rendered views is about 3.5 dB. Note also how the distortion due to depth information depends on the selected viewpoint and is higher for viewpoint 1 which is farther from the available view.

The second group of columns of Figure 19 instead shows the case where depth is assumed to be correctly transmitted and analyzes texture data transmission. Again the loss of the first packet does not allow image decoding and it is not shown in the plots. Due to the different organizations and meanings of the compressed data, even with a redundancy of 4 packets (out of 77 data packets, corresponding to a 5% of redundancy information) it is possible to obtain a 2 dB gain over the protection of the first packet. By doubling the number of redundancy packets the MSE is also decreased by a factor of 2. The biggest difference with the previous case is that here distortion is almost independent of the selected viewpoint.

Until this point depth and texture losses have been considered separately but one of the main targets of the proposed scheme is the allocation of the redundancy between the two kinds of data. Figure 20 concerns the case where packet loss affects both texture and geometry and the redundancy must be distributed between the two types of information. The right-most column shows the results obtained by protecting just the first packet while the other two correspond to 8 total packets of redundancy (for $62 + 77 = 139$ packets of source data) and to 16 redundancy packets. Again as expected, the MSE decreases as redundancy increases (from 15 if we protect just the first packet to 12 for $R = 8$ and to 6.5 for $R = 16$). The results depend on the selected view but not as strongly as in the depth case (the distortion in the rendered views shown in the figure is due both to texture distortion, roughly independent from the view, and to the distortion caused by depth losses which instead is view-dependent). An interesting observation is that in this case the distribution of R between texture and

geometry is not fixed but depends on the selected viewpoint, as explained in Section 3. Figure 21 shows the distribution of $R = 16$ redundancy packets between the various quality layers of depth and texture information: in the case of viewpoint 3 which is quite close (and so it does not require accurate depth data) we have 13 redundancy packets on texture (shown in various tones of blue) and just 3 for depth (green tones), while for the farther viewpoint 1 redundancy is almost equally distributed between the two types of data. To give an idea of the usefulness of this approach, the average MSE on the rendered frames from viewpoint 3 is 2.9 while it would be 4 if the redundancy would have been divided in half between the two elements of the description.

Figure 22 refers to the same set of experiments of the previous plot on a network with a 5% packet loss. Results are similar to the previous case but it can be observed that when transmitting depth information the FEC scheme with 5% of redundancy (4 packets) in this case works better, because the lower error rate permits an easier recovery of the lost packets.

Finally Figure 23 summarizes the previous results for the case of viewpoint 1. It shows the distortion as a function of the packet loss rate for the viewpoint corresponding to camera 1. As expected the distortion increases with the loss rate; however the plot clearly shows how the previously shown results for a 10% loss rate remain valid also in the case of lower loss rates. Table 1 shows also the distribution of the redundancy packets between the various quality layers in texture transmission for the different loss rates while Table 2 shows the same data for depth information. As previously said the first quality layers have a much bigger protection. In analyzing the data in the tables it should be noted that the lower layers have less packets, and so an equal amount of redundancy corresponds to higher protections. It should also be noted how at high loss rates almost all of the redundancy must be allocated to the first layers while in the case of a more reliable network some redundancy is allocated also to less relevant data.

4.4. Time Analysis of 3D Transmission with the Proposed FEC Protection Scheme. As shown in the previous section FEC protection schemes allow to obtain better quality. However they also require a longer transmission time due to the



(a)



(b)



(c)



(d)

FIGURE 14: *Breakdance*: (a) first frame of camera 4, (b) warping of the frame to the viewpoint of camera 3, (c) first frame of camera 4 corrupted by texture packet losses, (d) warping of the corrupted frame to viewpoint 3.

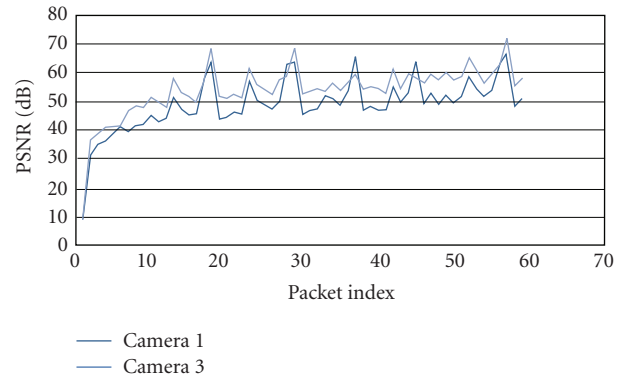


FIGURE 15: *Breakdance*: loss of a single packet of depth information.

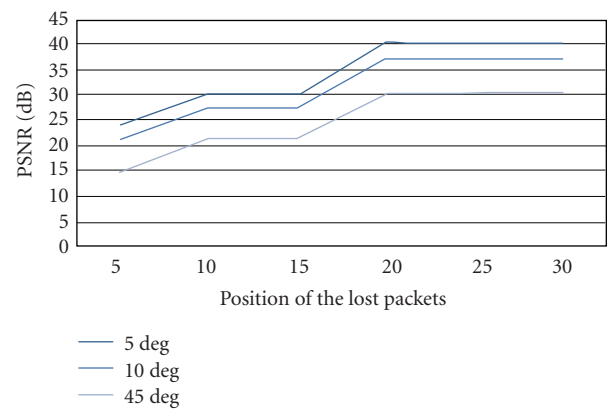
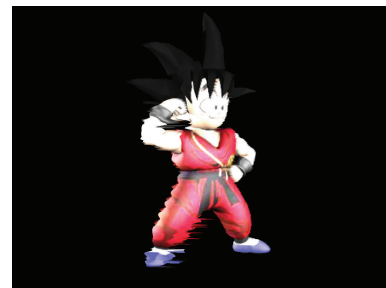
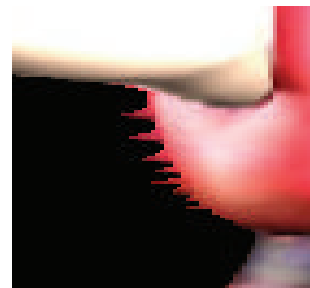


FIGURE 16: *Goku*: loss of a batch of 5 consecutive packets of depth information.

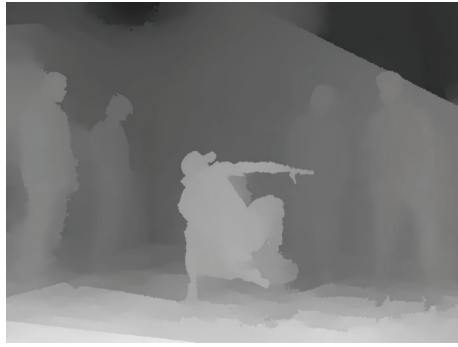


(a)



(b)

FIGURE 17: Example of artefacts due to depth packets loss.



(a)



(b)



(c)



(d)

FIGURE 18: *Breakdance*: (a) depth map from camera 4, (b) depth map from camera 4 corrupted by depth packet losses, (c) warping of the corresponding frame to viewpoint 3 using the corrupted depth, (d) warping of the corresponding frame to viewpoint 1 using the corrupted depth.

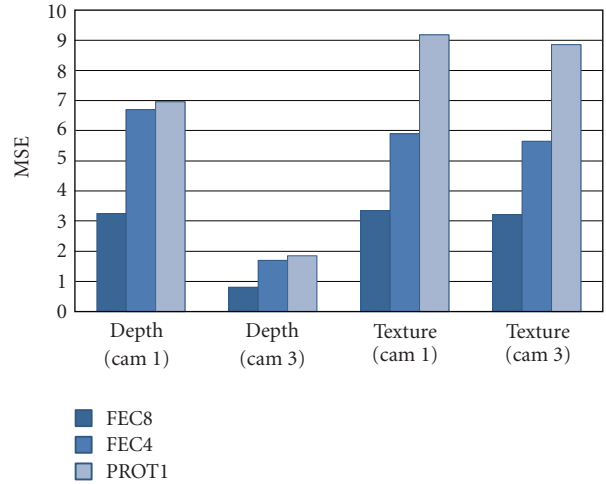


FIGURE 19: Performance at 10% packet loss rate: distortion due to depth (columns 1 and 2) and texture (columns 3 and 4) packet losses with different error protection schemes.

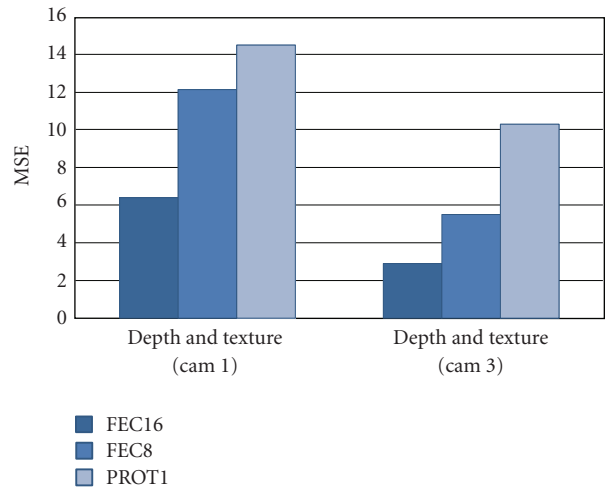


FIGURE 20: Performance with 10% packet loss rate on both texture and geometry: distortion due to packet losses with different error protection schemes.

overhead caused by redundancy information. In this section we will show an example of the quality versus latency trade off. Figure 24 refers to a network with a 10% packet loss and a FEC scheme with $R = 16$ redundancy packets. It shows the distortion as a function of the amount of transmitted data. Redundancy packets are concentrated on the first (more relevant) layers, so their transmission requires longer times, and at the very beginning the image quality is lower with the proposed FEC scheme (as shown by the zoomed detail of the plot). However it grows up faster and after a short while it gets better. Notice how average quality continues to grow with the proposed scheme, while unrecovered losses on early packets dominate the average quality without the protection scheme and cause the average quality curve to almost saturate.

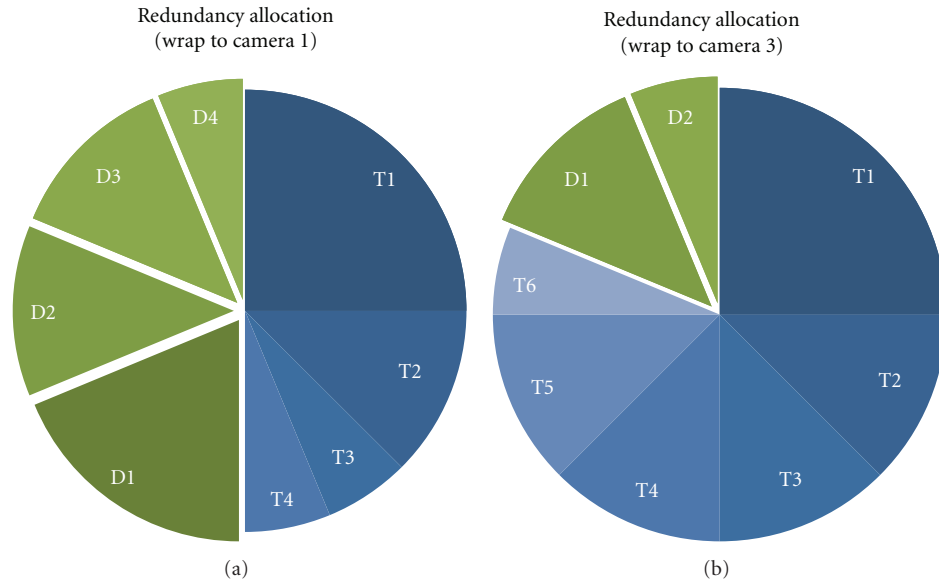


FIGURE 21: Allocation of the redundancy information between the various quality layers of texture (blue tones) and depth (green tones) information when observing from the viewpoints of Cam1 and Cam3.

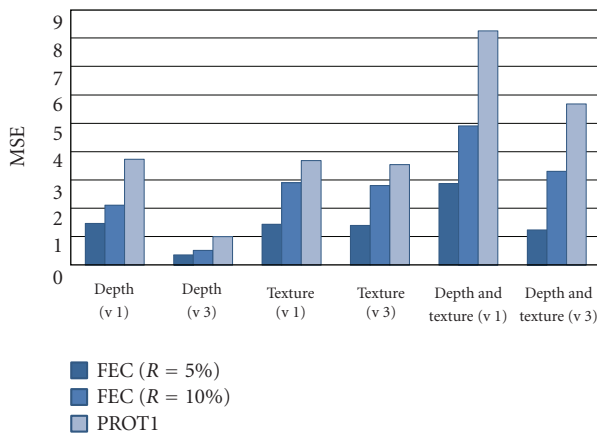


FIGURE 22: Performance at 5% packet loss rate: distortion due to depth (columns 1 and 2), texture (columns 3 and 4), and joint depth and texture (columns 5 and 6) packet losses with different error protection schemes.

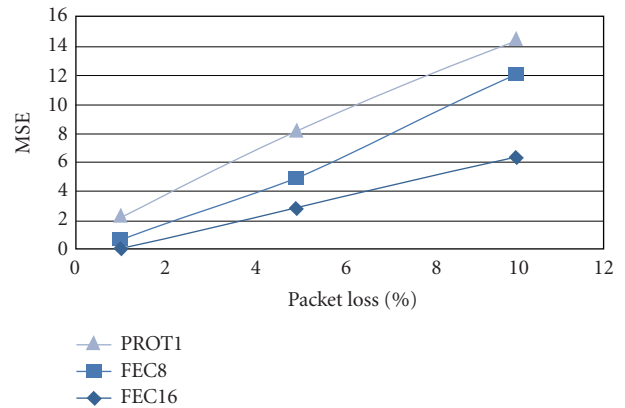


FIGURE 23: Loss of both depth and texture packets: distortion as a function of the packet loss rate with different error protection schemes (warping to viewpoint v1).

5. Conclusions and Future Work

In this paper we propose a novel error correction scheme for the combined transmission of geometry and texture information in depth image-based rendering schemes. The contributions of this paper are several. A first contribution is a novel Forward Error Correction strategy based on Unequal Error Protection that assigns the redundancy to the various elements of depth and texture information on the basis of their relevance and of the scene’s geometry and selected viewpoints. The proposed scheme has been tested in the transmission of three-dimensional scenes and experimental results show a considerable improvement in the actual rendering quality over lossy networks. This indicates that the

proposed method for assessing the relevance of the different depth and texture elements on the basis of the rendered views’ quality is rather effective for practical purposes. A second contribution is a model that theoretically describes the impact of the different texture and geometry elements on the rendering of novel views from arbitrary viewpoints. This model allows to estimate the effects of packet losses in the transmission of compressed depth and texture information in a remote 3D browsing system. Different approximation strategies have been proposed in order to cut a trade off between the accuracy and the computational requirements. The approximate computation can play valuable services in real-time applications, specially whenever distortion may be evaluated with limited accuracy, as when it is used to

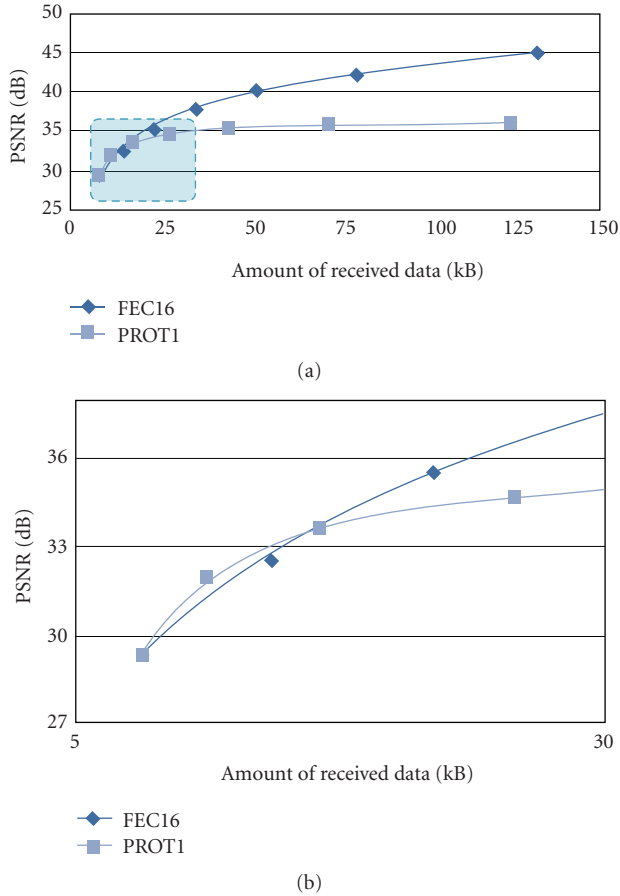


FIGURE 24: (a) Rendering distortion as a function of the amount of transmitted data (10% packet loss, warping to viewpoint v3); (b) Enlarged image of the highlighted region.

TABLE 1: Redundancy allocation for texture information ($R = 8$).

Quality layer	1	2	3	4	5	6	7	Tot
Data packets	2	2	3	6	10	19	35	77
Red. packets (1% loss)	2	1	1	1	1	1	1	8
Red. packets (5% loss)	3	1	1	1	1	1	0	8
Red. packets (10% loss)	4	2	1	1	0	0	0	8

balance redundancy between geometry and texture data. Experimental results confirm the theoretical findings and show that while the distortion due to the loss of texture packets is roughly independent of the selected viewpoint, the impact of loss of depth data becomes bigger and bigger while the viewpoints move farther apart from the one of the available images. The current version of the model estimates the MSE of the rendered views and uses this measure as an

TABLE 2: Redundancy allocation for depth information ($R = 8$).

Quality layer	1	2	3	4	5	6	7	Tot
Data packets	1	2	4	6	10	14	25	62
Red. packets (1% loss)	2	1	1	1	1	1	1	8
Red. packets (5% loss)	3	2	1	1	1	0	0	8
Red. packets (10% loss)	3	2	2	1	0	0	0	8

index of the image quality. Further research will be devoted to the introduction of more accurate and up-to-date metrics into the quality estimation model. Also the critical issue of how to combine image distortion due to depth and texture losses will be the subject of further research, and a more accurate model than the one presented in Section 2.4 will be developed and introduced in the system.

Further research will also focus on the interactivity issue with the target of efficiently applying the proposed scheme in free viewpoint video and 3DTV applications. While the experimental results have been obtained with a JPEG2000/JPIP remote browsing system, the proposed method applies to any compression scheme and its performance with other compression standards will be tested, with a special attention to video compression. The model for the estimation of the impact of depth losses will be improved and extended in order to deal with multiple images and depth maps. This aspect introduces very challenging new issues related to the possibility of replacing lost information from one view or depth map with data coming from other available viewpoints. We finally plan to reinforce the redundancy allocation procedure with a more accurate modeling of the dependency between the various data packets.

References

- [1] A. Smolic, K. Mueller, P. Merkle, et al., "3D video and free viewpoint video—technologies, applications and MPEG standards," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '06)*, pp. 2161–2164, July 2006.
- [2] S. Yang, C.-S. Kim, and C.-C. J. Kuo, "A progressive view-dependent technique for interactive 3D mesh transmission," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, pp. 1249–1264, 2004.
- [3] D. Tian and G. AlRegib, "On-demand transmission of 3D models over lossy networks," *Signal Processing: Image Communication*, vol. 21, no. 5, pp. 396–415, 2006.
- [4] N.-H. Lin, T.-H. Huang, and B.-Y. Chen, "3D model streaming based on JPEG 2000," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 1, pp. 182–190, 2007.
- [5] P. Zanuttigh, N. Brusco, D. Taubman, and G. Cortelazzo, "A novel framework for the interactive transmission of 3D

- scenes,” *Signal Processing: Image Communication*, vol. 21, no. 9, pp. 787–811, 2006.
- [6] S. Yang, C.-H. Lee, and C.-C. J. Kuo, “Optimized mesh and texture multiplexing for progressive textured model transmission,” in *Proceedings of the 12th Annual ACM International Conference on Multimedia (MULTIMEDIA '04)*, pp. 676–683, ACM, New York, NY, USA, 2004.
- [7] I. Cheng and A. Basu, “Perceptually optimized 3D transmission over wireless networks,” in *Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '05)*, Los Angeles, Calif, USA, August 2005.
- [8] F. Wu, E. Agu, C. Lindsay, and C. Chen, “Unequal error protection (uep) for wavelet-based wireless 3D mesh transmission,” in *Proceedings of the 8th IEEE International Symposium on Network Computing and Applications (NCA '09)*, pp. 242–249, Cambridge, Mass, USA, July 2009.
- [9] Z. Chen, J. F. Barnes, and B. Bodenheimer, “Hybrid and forward error correction transmission techniques for unreliable transport of 3D geometry,” *Multimedia Systems*, vol. 10, no. 3, pp. 230–244, 2005.
- [10] G. AlRegib, Y. Altunbasak, and J. Rossignac, “An unequal error protection method for progressively transmitted 3D models,” *IEEE Transactions on Multimedia*, vol. 7, no. 4, pp. 766–776, 2005.
- [11] J. Thie and D. Taubman, “Optimal erasure protection strategy for scalably compressed data with tree-structured dependencies,” *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2002–2011, 2005.
- [12] I. Cheng, L. Ying, and A. Basu, “Packet-loss modeling for perceptually optimized 3D transmission,” *Advances in Multimedia*, vol. 2007, no. 1, Article ID 95218, 10 pages, 2007.
- [13] A. Bilgin, Z. Wu, and M. W. Marcellin, “Decompression of corrupt jpeg 2000 codestreams,” in *Proceedings of the Data Compression Conference (DCC '03)*, Snowbird, Utah, USA, March 2003.
- [14] F. Dufaux, G. Baruffa, F. Frescura, and D. Nicholson, “JPWL— an extension of JPEG 2000 for wireless imaging,” in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '06)*, pp. 3870–3873, 2006.
- [15] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, 1993.
- [16] S. Floyd and K. Fall, “Promoting the use of end-to-end congestion control in the internet,” *IEEE/ACM Transactions on Networking*, vol. 7, no. 4, pp. 458–472, 1999.
- [17] L. Rizzo, “Effective erasure codes for reliable computer communication protocols,” *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, 1997.
- [18] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 2003.
- [19] Y. Pan, I. Cheng, and A. Basu, “Quality metric for approximating subjective evaluation of 3D objects,” *IEEE Transactions on Multimedia*, vol. 7, no. 2, pp. 269–279, 2005.
- [20] D. S. Taubman, “Localized distortion estimation from already compressed jpeg2000 images,” in *Proceedings of the IEEE International Conference of Image Processing (ICIP '06)*, pp. 3089–3092, Atlanta, Ga, USA, October 2006.
- [21] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, London, UK, 2nd edition, 2004.
- [22] A. Secker and D. Taubman, “Highly scalable video compression with scalable motion coding,” *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1029–1041, 2004.
- [23] P. Zanuttigh, N. Brusco, D. Taubman, and G. M. Cortelazzo, “Server policies for interactive transmission of 3D scenes,” in *Proceedings of the 8th IEEE International Workshop on Multimedia Signal Processing (MMSP '06)*, pp. 59–64, Victoria, Canada, October 2006.
- [24] D. Taubman and R. Prandolini, “Architecture, philosophy and performance of JPIP: internet protocol standard for JPEG 2000,” in *The International Symposium on Visual Communications and Image Processing (VCIP '03)*, vol. 5150 of *Proceedings of SPIE*, pp. 649–663, Lugano, Switzerland, July 2003.
- [25] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “Highquality video view interpolation using a layered representation,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 600–608, 2004.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

