Server Policies For Interactive Transmission Of 3D Scenes

Pietro Zanuttigh and Nicola Brusco
Università di Padova, Padova, Italy
Univ. of New South Wales, Sydney, Australia
Email: pietro.zanuttigh@dei.unipd.it
brusco@dei.unipd.it

David Taubman
University of New South Wales
Sydney, Australia
Email: d.taubman@unsw.edu.au

Guido Maria Cortelazzo
Università di Padova
Padova, Italy
Email: corte@dei.unipd.it

***Abstract*— We consider an interactive client-server application for remote browsing of 3D scenes. The information about texture and geometry is available at server side in the form of scalably compressed images and depthmaps, corresponding to a multitude of original image views. Image and depth components are both open to augmentation as more content becomes available. During the interactive browsing experience, the server allocates the available bandwidth between the delivery of new elements from the various original view bit-streams and new elements from the original geometry bit-streams. We propose a rate-distortion criterion to decide the best transmission policy for the server, since the best solution is not always to send the nearest original view image to the one which the client is rendering. We also outline how the JPIP standard for interactive transmission of JPEG2000 images can be exploited for remote exploration of 3D scenes.**

I. Introduction

This paper is concerned with the problem of efficient interactive transmission and rendering of 3D scenes. We envisage a server and a client, connected via a bandlimited channel. At the client side, a user interactively determines the view of interest, while the server delivers incremental contributions from two types of pre-existing data: scalably compressed images of the scene from a collection of pre-defined views V^i and a scalably compressed representation of the scene surface geometry. The user's interest may remain focused on a single view or change rapidly between different ones. The server should transmit the more useful views and geometry information while the client should exploit the received data to show the view of interest.

Considering the above arguments, we propose a framework for interactive scene browsing, in which the server uses a rate-distortion criterion to decide which set of data needs to be sent. Two big issues must be solved to exploit this approach: The first is how the client should combine information from available original view images into a new view of interest, using the avaiable description of the surface geometry. The second is how the server should distribute available transmission bandwidth amongst the various views and the geometry.

In this paper we focus our attention on the second question, since the first were the subject of a previous work [1]. Included in this question is whether the server should transmit elements from a new view which is more closely aligned with the requested view, or refine nearby original views for which the client already has more data. Importantly, the server models the client's current cache contents and uses this information to steer its policy. This leads to significant differences in the rate-distortion framework proposed here, from that found in related works such as [2].

The proposed solution for the server depends on how it expects the client to use the received information, and so the distortion sensitive approach to view synthesis we developed is briefly recalled. We propose a solution for the distribution of available resources between the reinforcement of available images and the transmission of new ones in the case of a fixed geometry. Improvement of geometry information will be the subject of future work.

II. Proposed method

A. Distortion sensitive view synthesis

In this section we briefly recall the method we developed to combine together a set of available views of the 3D scene into the rendering for the required view V^* using the geometry information represented as a triangular mesh or by a set of scalably compressed depthmaps.

Every available view V^i is reprojected on the target view V^* using the geometric information. In this way we obtain a set of warped images, covering overlapping regions of the required rendering. Simple solutions to combine the information from multiple original view images such as averaging and sticthing the renderings $\mathcal{W}^i\left(V^i\right)$, tend to cause blurring or discontinuites due to imperfections in the geometry description or ligthing issue. A possible solution to these problems consists in performing stitching within a multiresolution framework, such as the discrete wavelet transform (DWT). Such an approach performs much more smoothing in the lower frequencies than in the higher ones. All the images used in the proposed system are also compressed in JPEG2000, which is based on the DWT decomposition.

The procedure starts by separately warping each source image. Each of the warped image $V^{i\rightarrow *} \triangleq \mathcal{W}^i\left(V^i\right)$ is then decomposed by a D level DWT into a low resolution base image LL_D and a collection of high-pass subbands $\mathrm{HL}_d^{i\rightarrow *}$, $\mathrm{LH}_d^{i\rightarrow *}$ and $\mathrm{HH}_d^{i\rightarrow *}$.

0-7803-9752-5/06/$20.00 ©2006 IEEE

Noting that the synthesis operator $\mathcal{S}$ is linear, the recursive synthesis procedure can be written as

$$\begin{aligned}\mathrm{LL}_d &= \mathcal{S}\left(\mathrm{LL}_{d+1}, \mathbf{0}, \mathbf{0}, \mathbf{0}\right) + \mathcal{S}\left(\mathbf{0}, \mathrm{HL}_{d+1}, \mathrm{LH}_{d+1}, \mathrm{HH}_{d+1}\right) \\ &= \mathcal{S}_{\mathrm{L}}\left(\mathrm{LL}_{d+1}\right) + \underbrace{\mathcal{S}_{\mathrm{H}}\left(\mathrm{HL}_{d+1}, \mathrm{LH}_{d+1}, \mathrm{HH}_{d+1}\right)}_{R_d}\end{aligned}$$

Here, $\mathcal{S}_{\mathrm{L}}$ and $\mathcal{S}_{\mathrm{H}}$ are the low- and high-pass portions of a single stage of DWT synthesis, and R_d denotes the "detail" image formed from the three high-pass subbands at decomposition level $d+1$.

Our multi-resolution stitching algorithm proceeds by separately stitching each resolution component to form

$$R_d^*\left[\mathbf{p}\right] = \sum_i \rho_d^i\left[\mathbf{p}\right] \cdot R_d^{i \rightarrow *}\left[\mathbf{p}\right], \;\; d = 0, 1, 2, \ldots, D \qquad (1)$$

Here $\rho_d^i\left[\mathbf{p}\right] \geq 0$ represents a set of blending weights for each location $\mathbf{p}$ of resolution component R_d, such that $\sum_i \rho_d^i\left[\mathbf{p}\right] = 1$. As far as possible, $\rho_d^i[\mathbf{p}]$ is set to 1 for a *best stitching source* V^i and 0 elsewhere, where the best stitching source may be determined on a sample-by-sample basis, or on the basis of triangles taken from a mesh model of the surface geometry, depending on the implementation. The final rendered view V^* is created by DWT synthesis from the R_d^*.

The central issue for the client is how to select the blending weights $\rho_d^i\left[\mathbf{p}\right]$. We developed an approach based on the minimization of the distortion in the final rendered image. The first source of distortion to be taken into account is the quantization error in the DWT samples introduced by image compression. The error in every sample in subband b of the image V^i finds its way into resolution component $R^{i \rightarrow *}$ of $V^{i \rightarrow *}$ through DWT synthesis, warping and further DWT analysis. The quantization error contribution to $R_d^{i \rightarrow *}\left[\mathbf{p}\right]$ can be approximated by

$$D_d^{i \rightarrow *}\left[\mathbf{p}\right] = \sum_b W_{b \rightarrow d}\left[\mathbf{p}\right] \cdot D_b^i\left[\left(\mathcal{W}_{b \rightarrow d}^i\right)^{-1}\left(\mathbf{p}\right)\right]$$

where $D_b^i\left[\mathbf{k}\right]$ denotes the mean squared distortion at location $\mathbf{k}$ in subband b of view V^i and the weights $W_{b \rightarrow d}\left[\mathbf{p}\right]$ depend upon the source subband b, the target resolution component d and the local expansion/contraction properties of the surface warping operator $\mathcal{W}^i$. We are using $\left(\mathcal{W}_{b \rightarrow d}^i\right)^{-1}$ for the operator which maps locations $\mathbf{p}$ in the warped resolution component R_d, back to the corresponding location $\mathbf{k} = \left(\mathcal{W}_{b \rightarrow d}^i\right)^{-1}\left(\mathbf{p}\right)$ in subband b of V^i. A more comprehenisve description of this framework, including also the effects of missing high frequency content when $\mathcal{W}^i$ is expansive, can be found in [1].

Another important source of distortion is the uncertainty in the geometry description. Where the geometry is unreliable, we can achieve better results choosing views closer to V^*. The error in the surface geometry translates into uncertainty in the warping operator which represents a translational uncertainty, studied previously in [3]. Its effect depends upon the power spectrum of the views being warped, which we model using local estimates of the energy in each subband. Illuminant-dependent effects such as shading and reflection, must also be taken into account. Their contribution grows with the angle between views V^* and V^i. Ignoring specularity, we expect this distortion term to be proportional to the signal power, which is again derived from local estimates of the energy in each subband. Combining these contributions with those due to compression noise, we obtain a distortion model of the form

$$D_d^{i \rightarrow *}\left[\mathbf{p}\right] = \Theta_d^{i \rightarrow *}\left[\mathbf{p}\right] + \sum_b W_{b \rightarrow d}\left[\mathbf{p}\right] \cdot D_b^i\left[\left(\mathcal{W}_{b \rightarrow d}^i\right)^{-1}\left(\mathbf{p}\right)\right] \qquad (2)$$

where $\Theta_d^{i \rightarrow *}\left[\mathbf{p}\right]$ depends on local subband energy estimates and properties of the view normals and local surface geometry, as developed in [1] and more comprehensively in [4].

In the simplest case, the client selects blending weight $\rho_d^i\left[\mathbf{p}\right]$ equal to 1 if $i = \mathrm{argmin}_{i'} D_d^{i' \rightarrow *}\left[\mathbf{p}\right]$ and 0 otherwise. In practice, we sometimes need to perform a smoother blending operation in the vicinity of holes, where regions in R_d^* might not be fully visible from any view V^i due to the spatial support of the DWT analysis and synthesis operators [4]. In any event, it is sufficient to appreciate here that the blending weights depend strongly on distortion estimates and that these depend both on geometry and distortion in the compressed source views.

B. Server policies

For simplicity at present, we consider only the problem of distributing transmission bandwidth amongst the various view images V^i, assuming that geometry is perfectly represented. These methods can later be generalized to distribute bandwidth between geometry and view data.

We build our client-server distribution policy on top of the JPIP communication [5] paradigm, which is ideally suited to the incremental dissemination of JPEG2000 content. Our source images V^i are all compressed using JPEG2000 and are thus represented by a collection of code-blocks $\mathcal{B}_\beta^i$, corresponding to different spatial regions of each subband of each view. JPIP servers deliver incremental contributions from these code-blocks in the form of *precinct data-bin byte ranges* [5], where a precinct corresponds to the set of co-located code-blocks from a given resolution component of the source view. Precincts used with JPIP typically have only three code-blocks each (one from each of the HL_d, LH_d and HH_d subbands). For this work, we assume that the JPIP server always delivers precinct data-bin contributions which correspond to a whole number of quality layers, q.

JPIP servers typically work in epochs; in each epoch, corresponding to a fixed time step and amount of data to be transmitted, the server creates a list of data-bin increments to be transmitted [5]. Let $D_{\beta,q}^i$ be the squared error distortion contributed to the decompressed image V^i by code-block $\mathcal{B}_\beta^i$ when only the first q quality layers are available. Also let $\left\{\mathcal{P}_\pi^i\right\}_\pi$ denote the collection of precincts from view V^i, indexed by π. Finally, we assume that the server keeps track of the number of quality layers q_π^i, which are available in the client's cache for all code-blocks in precinct $\mathcal{P}_\pi^i$ – this is normal for JPIP servers.

We are now in a position to describe the server's optimization objective. Since our spatial multi-resolution transform is based on an approximately orthonormal DWT, the total distortion associated with the reconstructed view V^* can be expressed as $D^* \approx \sum_{d,\mathbf{p}} D_d^{i\rightarrow *}[\mathbf{p}]$. Combining this with equations (1) and (2), we find

$$D^* \approx \sum_d \sum_{\mathbf{p}} \sum_i \left(\rho_d^i[\mathbf{p}]\right)^2 \cdot \left(\Theta_d^{i\rightarrow *}[\mathbf{p}] + \sum_b W_{b\rightarrow d}^i[\mathbf{p}] \cdot D_b^i\left[\left(\mathcal{W}_{b\rightarrow d}^i\right)^{-1}(\mathbf{p})\right]\right) \quad (3)$$

where $\Theta_d^{i\rightarrow *}[\mathbf{p}]$ represents the collective contribution of the illuminant- and geometry-dependent distortion terms appearing in equation (2). Contribution of ρ_d^i here is squared since we are computing distortion as the mean square error of subbands samples, obtained from equation 1.

The server's objective is to deliver precinct data-bin packets from the various view images in such a way as to minimize the objective D^* in equation (3), subject to a constraint $L \leq L^{\text{epoch}}$ on the amount of data to be transmitted in the next epoch. This problem can be reformulated in Lagrangian fashion as a family of unconstrained optimization objectives $J^*(\lambda)$, parametrized by $\lambda > 0$, where

$$J^*(\lambda) = D^*(\lambda) + \lambda L(\lambda). \quad (4)$$

The minimizing solution to each $J^*(\lambda)$ has the obvious property that $D^*(\lambda)$ is as small as it can be without increasing L beyond $L(\lambda)$. As a result, our global constrained optimization problem is solved once we find λ such that $L(\lambda) = L^{\text{epoch}}$. The discrete nature of the problem means that we can rarely find such a λ. In practice, therefore, noting that $L(\lambda)$ is a decreasing function of λ, we select the smallest λ for which $L(\lambda) \leq L^{\text{epoch}}$.

The optimization problem expressed by equations (3) and (4) is complicated by the fact that the blending weights $\rho_d^i[\mathbf{p}]$ themselves depend upon the local distortion in the view images. To address this difficulty, the server considers two sets of blending weights: $\mathring{\rho}_d^i[\mathbf{p}]$ denotes the blending weights which the client is assumed to be using at present, while $\vec{\rho}_d^i[\mathbf{p}]$ denotes the blending weights which would be used by the client if all source views V^i were available with maximum available quality. The server considers two types of enhancement to the client's existing cache contents, which we identify as *"reinforcing enhancements"* and *"disruptive enhancements."*

C. Optimization of reinforcing enhancements

Reinforcing enhancements are based on the assumption that the blending weights will not change between this epoch and the next. In this case the server will tend to send more information for those code-blocks which contribute most strongly to the client's view synthesis process. In this situation the terms $\Theta_d^{i\rightarrow *}[\mathbf{p}]$ in equation (3) represent a constant offset to D^*, which we can ignore. Our optimization objective is thus to minimize

$$\mathring{J}^*(\lambda) = \lambda L(\lambda) + \sum_i \sum_b \sum_{\mathbf{k}} \Psi_b^i[\mathbf{k}] \cdot D_b^i[\mathbf{k}]$$

where

$$\mathring{\Psi}_b^i[\mathbf{k}] = \sum_{d,\mathbf{p}\ni\left(\mathcal{W}_{b\rightarrow d}^i\right)^{-1}(\mathbf{p})=\mathbf{k}} \left(\mathring{\rho}_d^i[\mathbf{p}]\right)^2 W_{b\rightarrow d}^i[\mathbf{p}]$$

As mentioned above, the server is assumed to know only the total squared error distortion $D_{\beta,q}^i$ associated with each code-block $\mathcal{B}_\beta^i$ when its representation is truncated beyond quality layer q. In the absence of more precise information, therefore, we model the distortion at each sample $\mathbf{k}$ in code-block $\mathcal{B}_\beta^i$ as

$$D_{b(\beta)}^i[\mathbf{k}] = D_{\beta,q}^i / \left|\mathcal{B}_\beta^i\right|$$

where $\left|\mathcal{B}_\beta^i\right|$ is the number of samples in $\mathcal{B}_\beta^i$ and $b(\beta)$ denotes the subband to which code-block $\mathcal{B}_\beta^i$ belongs. With this model, we can express our objective in terms of precinct data-bin decisions, as

$$\mathring{J}^*(\lambda) = \sum_i \sum_\pi \left[\lambda\left(L_{\pi,q_\pi^i}^i - L_{\pi,\mathring{q}_\pi^i}^i\right) + \sum_{\mathcal{B}_\beta^i \in \mathcal{P}_\pi^i} \frac{\mathring{\Psi}_\beta^i}{\left|\mathcal{B}_\beta^i\right|} \cdot D_{\beta,q_\pi^i}^i\right] \quad (5)$$

where $L_{\pi,q}^i$ is the number of precinct data-bin bytes associated with the first q packets in precinct $\mathcal{P}_\pi^i$, q_π^i is the total number of packets (or quality layers) which the client will have received for precinct $\mathcal{P}_\pi^i$ by the end of this epoch, $\mathring{q}_\pi^i$ is the number of packets which the client already had in its cache for precinct $\mathcal{P}_\pi^i$ at the beginning of this epoch, and

$$\mathring{\Psi}_\beta^i = \sum_{\mathbf{k}\in\mathcal{B}_\beta^i} \mathring{\Psi}_{b(\beta)}^i[\mathbf{k}]$$

Evidently equation (5) decomposes into a set of independent optimization objectives for each precinct, for which the optimal solution is given by

$$q_\pi^i(\lambda) = \underset{q_\pi^i \geq \mathring{q}_\pi^i}{\text{argmin}} \left[\lambda L_{\pi,q_\pi^i}^i + \underbrace{\sum_{\mathcal{B}_\beta^i \in \mathcal{P}_\pi^i} \frac{\mathring{\Psi}_\beta^i}{\left|\mathcal{B}_\beta^i\right|} \cdot D_{\beta,q_\pi^i}^i}_{\mathring{D}_{\pi,q}^i}\right] \quad (6)$$

The server can easily solve this problem, using its knowledge of $D_{\beta,q_\pi^i}^i$ and $L_{\pi,q_\pi^i}^i$.

D. Optimization of disruptive enhancements

The optimization of disruptive enhancements is still based on the distortion model in equation (3). We begin by considering the overall policy switching penalty, associated with a wholesale change of all blending weights from $\mathring{\rho}_d^i[\mathbf{p}]$ to $\vec{\rho}_d^i[\mathbf{p}]$.

This may be written as

$$\Delta D^* = \sum_{d,\mathbf{p}} \underbrace{\sum_i \begin{array}{l}\left[\left(\vec{\rho}_d^{\,i}[\mathbf{p}]\right)^2 - \left(\mathring{\rho}_d^{\,i}[\mathbf{p}]\right)^2\right] \times \\ \left(\Theta_d^{i\rightarrow *}[\mathbf{p}] + \sum_b W_{b\rightarrow d}^i[\mathbf{p}] \cdot D_b^i\left[\left(\mathcal{W}_{b\rightarrow d}^i\right)^{-1}(\mathbf{p})\right]\right)\end{array}}_{\Phi_d[\mathbf{p}]} \tag{7}$$

where the subband distortions $D_b^i[\mathbf{k}]$ are assessed using the current number of layers $\mathring{q}_\pi^i$ for each precinct data-bin $\mathcal{P}_\pi^i$. The key step is to distribute this penalty amongst the various source view precincts. There is no perfect way to do this, since blending weights are assigned locally within the resolution components of V^*, rather than the precincts of V^i. Nevertheless, we argue that the most appropriate way to distribute ΔD^* is on the basis of the distortion weights, $\left(\vec{\rho}_d^{\,i}[\mathbf{p}]\right)^2 W_{b\rightarrow d}^i[\mathbf{p}]$, which would apply if the policy switch took place. This associates the policy switching penalty with those source precincts whose distortion impacts the synthesized view most, and these are the ones for which disruptive enhancements are likely to be sent, if at all. [1]

Specifically, we first observe that ΔD^* can be rewritten as

$$\begin{aligned}\Delta D^* &= \sum_d \sum_{\mathbf{p}} \Phi_d[\mathbf{p}] = \sum_d \sum_{\mathbf{p}} \sum_i \sum_b \Phi_{d,b}^i[\mathbf{p}] \\ &= \sum_i \sum_\pi \Phi_\pi^i\end{aligned}$$

where

$$\Phi_{d,b}^i[\mathbf{p}] = \Phi_d[\mathbf{p}] \cdot \frac{\left(\vec{\rho}_d^{\,i}[\mathbf{p}]\right)^2 W_{b\rightarrow d}^i[\mathbf{p}]}{\sum_{b',i'}\left(\vec{\rho}_{d'}^{\,i'}[\mathbf{p}]\right)^2 W_{b'\rightarrow d}^{i'}[\mathbf{p}]}$$

and

$$\Phi_\pi^i = \sum_{\beta \in \mathcal{P}_\pi^i} \sum_{d,\mathbf{p} \ni \left(\mathcal{W}_{b(\beta)\rightarrow d}^i\right)^{-1}(\mathbf{p}) \in \mathcal{B}_\beta^i} \Phi_{d,b}^i[\mathbf{p}]$$

Although these equations may look complex, they simply represent the steps of estimating distortion in each resolution component of the view image, distributing it back to the source view subbands using the same weights that are used while calculating distortion, except that the weights used to calculate ΔD^* are $\left(\left(\vec{\rho}_d^{\,i}[\mathbf{p}]\right)^2 - \left(\mathring{\rho}_d^{\,i}[\mathbf{p}]\right)^2\right) W_{b\rightarrow d}^i$, whereas the weights used to distribute it back to the precincts are $\left(\vec{\rho}_d^{\,i}[\mathbf{p}]\right)^2 W_{b\rightarrow d}^i$. These steps can readily be performed together in a software implementation. Also, the computation in the server can be performed at reduced resolution, since decisions in the end must be made for whole code-blocks.

The terms Φ_π^i represent the policy switching penalties associated with each precinct $\mathcal{P}_\pi^i$. We consider this contribution to ΔD^* to be incurred if and only if disruptive enhancements for $\mathcal{P}_\pi^i$ are delivered. Of course, this is not strictly correct, since the client's blending policy depends on the impact of multiple subbands. However, if disruptive enhancement proves worthwhile for one contributing precinct, it is likely also to prove worthwhile for the other main contributors, either in this epoch or a subsequent one.

Let us suppose that such a disruptive enhancement provides q quality layers for precinct $\mathcal{P}_\pi^i$, at a cost of $L_{\pi,q}^i - L_{\pi,\mathring{q}_\pi^i}^i$ bytes. These bytes reduce the distortion D^* by an amount $\vec{D}_{\pi,\mathring{q}_\pi^i}^i - \vec{D}_{\pi,q}^i$, where

$$\vec{D}_{\pi,q}^i = \sum_{\mathcal{B}_\beta^i \in \mathcal{P}_\pi^i} \vec{\Psi}_\beta^i \cdot D_{\beta,q}^i, \quad \vec{\Psi}_\beta^i = \sum_{\mathbf{k} \in \mathcal{B}_\beta^i} \vec{\Psi}_{b(\beta)}^i[\mathbf{k}] \quad \text{and}$$

$$\vec{\Psi}_b^i[\mathbf{k}] = \sum_{d,\mathbf{p} \ni \left(\mathcal{W}_{b\rightarrow d}^i\right)^{-1}(\mathbf{p}) = \mathbf{k}} \left(\vec{\rho}_d^{\,i}[\mathbf{p}]\right)^2 W_{b\rightarrow d}^i[\mathbf{p}].$$

These definitions for $\vec{D}_{\pi,q}^i$ and $\vec{\Psi}_\beta^i$ are, of course, identical to those for $\mathring{D}_{\pi,q}^i$ and $\mathring{\Psi}_\beta^i$, used when considering reinforcing enhancements, except that $\mathring{\rho}_d^{\,i}[\mathbf{p}]$ is replaced by $\vec{\rho}_d^{\,i}[\mathbf{p}]$.

Combining the length and distortion contributions, we see that the impact on $J^*(\lambda)$ of a disruptive enhancement which assigns q_π^i layers to precinct $\mathcal{P}_\pi^i$ is

$$\vec{J}_{\pi,q_\pi^i}^{\,i}(\lambda) = \begin{cases} 0, & q_\pi^i = \mathring{q}_\pi^i \\ \Phi_\pi^i + \left(\vec{D}_{\pi,q_\pi^i}^i - \vec{D}_{\pi,\mathring{q}_\pi^i}^i\right) + & \\ \lambda\left(L_{\pi,q_\pi^i}^i - L_{\pi,\mathring{q}_\pi^i}^i\right), & \mathring{q}_\pi^i < q_\pi^i \end{cases}$$

The above expression can be considered valid only if $q_\pi^i > \mathring{q}_\pi^i$ is sufficiently large to ensure that $\Phi_\pi^i + \left(\vec{D}_{\pi,q_\pi^i}^i - \vec{D}_{\pi,\mathring{q}_\pi^i}^i\right) < 0$. This is the condition under which disruptive enhancement is expected to cause policy switching. Otherwise, any enhancement is expected to be of the reinforcing type, discussed in the previous sub-section.

It is instructive to consider solutions to the problem

$$q_\pi^i(\lambda) = \underset{q_\pi^i \geq \mathring{q}_\pi^i}{\operatorname{argmin}} \vec{J}_{\pi,q_\pi^i}^{\,i}(\lambda) \tag{8}$$

These may also be interpreted as solutions to the problem

$$q_\pi^i(\lambda) = \underset{q_\pi^i \geq \mathring{q}_\pi^i}{\operatorname{argmin}} \left[\lambda L_{\pi,q_\pi^i}^i(\lambda) + \bar{D}_{\pi,q_\pi^i}^i\right]$$

where

$$\bar{D}_{\pi,q}^i = \begin{cases} \vec{D}_{\pi,\mathring{q}_\pi^i}^i - \Phi_\pi^i, & q = \mathring{q}_\pi^i \\ \vec{D}_{\pi,q}^i, & q > \mathring{q}_\pi^i \end{cases}$$

A typical plot of the $\bar{D}_{\pi,q}^i$ vs. $L_{\pi,q}^i$ characteristic is shown in Figure 1.

Let $\vec{\mathcal{H}}_\pi^i$ denote the convex hull of the $\left(L_{\pi,q}^i, \bar{D}_{\pi,q}^i\right)$ pairs. This is identified in Figure 1 as the *effective convex hull*. Only those q which belong to $\vec{\mathcal{H}}_\pi^i$ can arise as solutions to equation (8). All points $q > \mathring{q}_\pi^i$ which belong to $\vec{\mathcal{H}}_\pi^i$ must necessarily satisfy the condition $\bar{D}_{\pi,q}^i < \bar{D}_{\pi,\mathring{q}_\pi^i}^i$, which is equivalent to $\Phi_\pi^i + \left(\vec{D}_{\pi,q_\pi^i}^i - \vec{D}_{\pi,\mathring{q}_\pi^i}^i\right) < 0$. As a result, any $q_\pi^i(\lambda) > \mathring{q}_\pi^i$ which arises as a solution to equation (8) already satisfies the condition required for the enhancement to be

[1] If contributions of precincts to the switching decision were completely disjoint, i.e. precints contributed to the switching of resolution components of V^* which do not overlap, this approach would be exact.

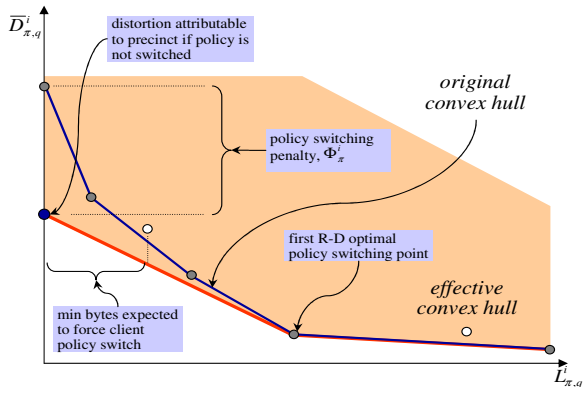


Fig. 1. *Effective distortion-length slope properties associated with disruptive enhancement.*

considered disruptive. As suggested by Figure 1, for large policy switching penalties Φ^i_π, the first $q > \mathring{q}^i_\pi$ which belongs to $\vec{\mathcal{H}}^i_\pi$ can lie significantly beyond the point at which $\bar{D}^i_{\pi,q} < \bar{D}^i_{\pi,\mathring{q}^i_\pi}$, which adds to our confidence that a policy switch will actually occur within the client synthesis procedure.

E. Complete server solution

To combine together reinforcing and disruptive enhacement we simply take the maximum of the $q^i_\pi(\lambda)$ values yielded by equations (6) and (8), i.e.

$$q^i_\pi(\lambda) = \max \left\{ \begin{array}{c} \mathrm{argmin}_{q^i_\pi \geq \mathring{q}^i_\pi} \left[\lambda L^i_{\pi,q^i_\pi}(\lambda) + \bar{D}^i_{\pi,q^i_\pi} \right], \\ \mathrm{argmin}_{q^i_\pi \geq \mathring{q}^i_\pi} \left[\lambda L^i_{\pi,q^i_\pi} + \mathring{D}^i_{\pi,q} \right] \end{array} \right\}.$$

An outer loop adjusts λ until $\sum_{i,\pi} L^i_\pi(\lambda) \approx L^{\text{epoch}}$, exploiting the fact that the solutions to (6) and (8) are both non-increasing functions of λ.

This "max of solutions" policy tends to send somewhat more reinforcing enhancements than might be involved in a truly optimal solution[2], which may be regarded as a conservative position. This is because we cannot be completely sure that disruptive enhancements will actually cause the anticipated policy shift, due to the mixing of many contributions in the policy decision represented by equation (2). In any event, blending weights will be recomputed in the next epoch, so that any sub-optimality represented by the "max of solutions" approach is limited by the size of the epoch.

III. Experimental results and discussion

In order to evaluate the performance of the proposed transmission approach, we show some results with the transmission of the 3D model of " Goku ". At the client side, two images V_L and V_R are available at 0.025 bpp on the left and right side of the 3D model, which is rendered from the front side. Three images (V_R, V_L and V_F, which is the front image) are available at full quality at the server side. The server is going to send data in epochs in order to improve the quality of the rendering. In each epoch the server can deliver up to 2048 bytes. They are shared among the three images V_L, V_R and V_F as shown in Figure 2, where the y axis represents the total bytes sent. At the beginning, no data is available for V_F. In epoch 1 and epoch 2, disruptive enhancement turns larger than enforcing one. However, not all the bytes are spent on V_F, even if it is best aligned with the viewer. This is because precincts of V_F will not decrease the distortion enough to compensate the policy switching penalty Φ^i_π. From epoch 3, the policy completely switches to V_F, which receives almost the totality of the bytes spent by the server. In Figure 2 the case when one only image V_F is sent is also shown: all bytes are obviously spent on it. After some epochs the server delivers the same data in the two cases (when only V_F is sent and when all V_L, V_R and V_F are delivered). A gap between the two cases is due to the bytes which were sent at the beginning for V_L, V_R. Those bytes allow to obtain an acceptable rendering in epoch 1 and 2 even if not many bytes are available for V_F yet. In Figure 3 the bytes allocation in the first three epochs, i.e. the additional bytes sent, is shown. The first group of columns of each epoch shows the bytes which would have been sent if only the enforcing decision was taken into account; the second one is about the disruptive decision only; the third one shows the combined solution. In epoch 1 no enforcing contribution is given to V_F, which has no data; most of disrupting contribution, instead, goes to V_F. In epoch 2 disrupting contribution is quite balanced; from epoch 3 all the contributions are allocated to V_F, since blending choices now discard V_L and V_R.

In Figure 4 visual evidence is provided, through a detail of the model in different epochs. The upper row shows the model when only V_F is sent; the lower row shows the rendering when all V_L, V_R and V_F are delivered. In the second epoch the visual result is much better with three views than with only one, since V_L and V_R already provide some useful data. From epoch 2, the server starts delivering almost all bytes for V_F in both cases, while V_L and V_R become useless and are discarded. PSNR curves do not give a reasonable representation of reality because illumination issues and view warping artifacts (translational shifts) due to imperfections in the geometry, as described in [1], introduce a large contribution to the MSE that doesn't correspond to the perceived visual quality of the image.

IV. Conclusions and future work

In this work we described a framework for interactive scene browsing, in which the server uses a rate-distortion criterion to decide which set of data needs to be sent, and proposed an effective solution for the allocation of the available bandwidth. Two steps are accomplished: an enforcing one and a disrupting one. The first one minimizes the distortion according to the blending choices based on the available data; the second one takes into account a disruptive change in the blending choices, due to new data transmitted to the client.

[2]An optimal solution should take into account the fact that disruptive enhancements could change the blending weigths, while reinforcing enhancements are based on the assumption that they don't change.

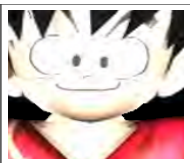
1 View	***No Data***					
3 Views						
Epochs	0	1	2	3	4	5

Fig. 4. *Detail of "Goku" in different epochs.*

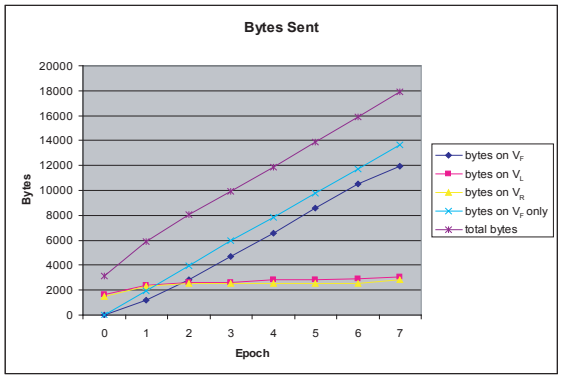


Fig. 2. *Total bytes sent by the server in different epochs, shared among images.*

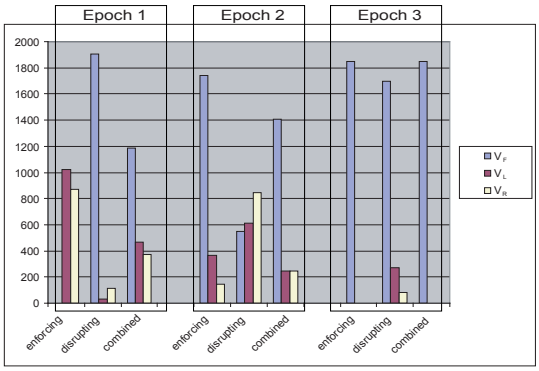


Fig. 3. *Bytes allocation between different views in the first three epochs.*

Disruptive element is of particular interest, since the solution to the R-D server optimization problem depends discontinuously on λ and hence on the available transmission budget. Thus, it represent a step towards the solution of the problem of non-linear approximation of the plenoptic function. This solution is based on the desired viewpoint, the transmission rate, the desired response time (through L^{epoch}) and the data already available at the client.

In this paper no progressive refinement of geometry has been considered. Current work aims to introduce in the framework the progressive transmission of geometric information, and to develop a combined distortion model considering also the influence of geometric error on the quality of the rendering. In this way it will be possible to find an optimal solution for the allocation of the available bandwidth between geometry and textures.

REFERENCES

[1] P. Zanuttigh, N. Brusco, D. Taubman, and G. Cortelazzo, "Greedy non-linear optimization of the plenoptic function for interactive transmission of 3d scenes," *International Conference of Image Processing ICIP2005, Genova*, September 2005.

[2] P. Ramanathan and B. Girod, "Receiver-driven rate-distortion optimized streaming of light fields," in *Icip*, vol. 3. IEEE, September 2005, pp. 25–28.

[3] D. Taubman and A. Secker, "Highly scalable video compression with scalable motion coding," *Proceedings of International Conference on Image Processing*, vol. 3, no. 3, pp. 273–276 v.2, September 2003.

[4] N. Brusco, P. Zanuttigh, D. Taubman, and G. M. Cortelazzo, "Distortion-sensitive synthesis of texture and geometry in interactive 3d visualization," in *Submitted on Int. Symp. on 3DPVT*, 2006.

[5] D.Taubman and R.Prandolini, "Architecture, philosophy and performance of jpip: internet protocol standard for JPEG 2000," in *Int. Symp. Visual Comm. and Image Proc.*, vol. 5150. IEEE, July 2003, pp. 649–663.