Available online at www.sciencedirect.com
ScienceDirect
Signal Processing: *Image Communication* 21 (2006) 787–811
SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

# A novel framework for the interactive transmission of 3D scenes

Pietro Zanuttigh[a,b], Nicola Brusco[a,b], David Taubman[b,*,1], Guido Cortelazzo[a]

[a]*University of Padua, Padua, Italy*
[b]*School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney, Australia*

Received 21 February 2006; received in revised form 7 June 2006; accepted 16 June 2006

**Abstract**

We consider an interactive browsing environment for 3D scenes, which allows for the dynamic optimization of selected client views by distributing available transmission resources between geometry and texture components. Texture information is available at a server in the form of scalably compressed images, corresponding to a multitude of original image views. Surface geometry is also available at the server in the form of scalably compressed depth maps, again corresponding to a multitude of original views. Texture and depth components are both open to augmentation as more content becomes available. At any point in the interactive browsing experience, the server must decide how to allocate transmission resources between the delivery of new elements from the various original view bit-streams and new elements from the original geometry bit-streams. The proposed framework implicitly supports dynamic view sub-sampling, based on rate-distortion criteria, since the best server policy is not always to send the nearest original view image to the one which the client is rendering. In this paper, we particularly elaborate upon a novel strategy for distortion-sensitive synthesis of both geometry and rendered imagery at the client, based upon whatever data is provided by the server. We also outline how the JPIP standard for interactive communication of JPEG2000 images, can be leveraged for the 3D scene browsing application.


*Keywords:* 3D Scene compression; Interactive browsing; Scene rendering; Wavelet transforms; JPEG 2000; Distortion modelling

## 1. Introduction

This paper is concerned with the problem of efficient interactive retrieval and rendering of 3D scene information. One basic issue addressed in some preliminary studies [4] is how transmission resources should be distributed between texture and geometry information. Current practical solutions for remote visualization of 3D scenes and models, however, somehow represent a limited theoretical understanding of this and related issues. One common approach involves the server transmitting a complete 3D model, together with its texture, to the remote client. Of course, such an approach suffers from poor response time, since visualization cannot commence until everything has been sent.

In the present paper, we envisage a server and a client, connected via a bandlimited channel. At the client side, a user interactively determines the particular view of interest. An important feature

*Corresponding author. Tel.: +61 2 9385 5223; fax: +61 2 9385 5993.
*E-mail address:* d.taubman@unsw.edu.au (D. Taubman).
[1]To be considered for the upcoming Special Issue on "Interactive Representation of Still and Dynamic Scenes".

0923-5965/$ - see front matter © 2006 Published by Elsevier B.V.
doi:10.1016/j.image.2006.06.003

of such applications is that the user can be expected to navigate between a variety of different views, although we do not know ahead of time which views will be of interest. We also do not know in advance how much time (transmission resources) the user will choose to devote to any particular view.

At one extreme, the user's interest may remain focused on a single view for a considerable period of time, waiting until very high quality imagery has been recovered before moving on. At this extreme, the interactive retrieval problem is tantamount to that of interactive image browsing, which is addressed most elegantly by progressive transmission of a single scalably compressed image, formed at the server. One way to achieve this is to combine the JPIP interactive imaging protocol with a JPEG2000 compressed representation of the view in question [17].

At the opposite extreme, the interactive user may select many different views in rapid succession, with the aim of understanding the scene's geometry. This phase might itself be a precursor to later detailed inspection of some particular view of interest. Since successive views are closely related, one natural way to improve the efficiency of the browsing experience is to predict each new view from the views which have already been transmitted, forming the same prediction at the server and client so that only the prediction residual need be transmitted. Explorations along this direction may be found in, e.g., [12,5,11]. The predictive approach, however, suffers from a number of drawbacks. Firstly, the server must precisely replicate the steps used by the client to render each new view from existing previous views. Secondly, the server must compress the residual images corresponding to each change of view by the client. Perhaps most importantly, the predictive approach delivers a distinct approximate representation for each view requested by the interactive user, no matter how close those views may be to each other. It is difficult, if not impossible, to combine the information from several similar yet-different lower-quality views to synthesize a new, higher-quality image at a later time. This limits the extent to which previously transmitted data can be leveraged in the future.

Considering the above arguments, we propose a framework for interactive scene browsing, in which the server delivers incremental contributions from two types of pre-existing data: (1) scalably compressed images of the scene from a collection of predefined views, $V^i$; and (2) a scalably compressed representation of the scene surface geometry, $G$. These elements are depicted in Fig. 1. We use the term "original view images" to distinguish the compressed server images $V^i$ from new views rendered by the client. The server does not generate new views or compress differential imagery. Instead, it determines and sends appropriate elements from a fixed set of scalable compressed bit-streams, so as to provide its clients with the most appropriate data from which to render their desired views.

Our proposed framework is particularly appropriate in view of the fact that 3D scene representations are usually generated from a collection of original 2D images; these are natural candidates for $V^i$. If the client happens to request one of the original view images, it can be incrementally served directly from its scalably compressed representation. Interestingly, though, this might not always be the best policy. If the client has already received sufficient elements (sufficient quality) from one or more nearby original view images, $V^{k_1}, V^{k_2}, \ldots$, it may be more efficient to send only the geometric information required for the client to synthesize the requested view, using the resulting bandwidth savings to further augment the quality of these nearby original view images. It follows that even if the server has a huge number of original view images, an efficient service policy would effectively subsample them based on the interactive user's navigation patterns. More generally, the server may

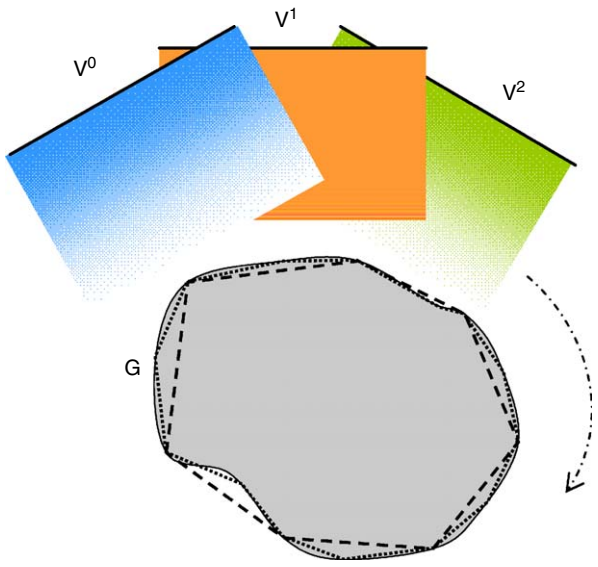


Fig. 1. Overview of the browsing environment. The server has scalably compressed representations for a set of "original view images," $V^i$ and surface geometry, $G$.

choose to send some elements from $V^i$, while expecting the client to derive other aspects of the view from the previously delivered, but less closely aligned original view images, $V^{k_n}$.

The proposed framework may thus be interpreted as fostering a greedy strategy for non-linear approximation of the plenoptic function, since it considers both view sub-sampling and rate-distortion criteria. The fact that efficient service policies can be expected to sub-sample the existing content automatically, brings the proposed approach into contrast with the predictive approach mentioned previously, where imagery is delivered for every view requested by the user. The system outlined above gives rise to the following interesting questions:

(1) How should the client combine information from available original view images into a new view of interest, using an available description of the surface geometry?
(2) How should the server distribute available transmission resources amongst the various original views and the geometry information which the client may need to render a new view? Included in this question is that of whether the server should transmit elements from a new original view which is more closely aligned with the requested view, rather than refining nearby original views for which the client already has more data.

Within the scope of this present paper, it is not possible to explore both of these questions in detail. Instead, we focus our attention on the first, since answers to the second question depend on how the server expects the client to use the information which it has. In particular, we develop further the paradigm we first presented in [19]. Apart from this previous work of our own, perhaps the most closely related ideas found in the literature are those of Ramanathan and Girod [13], who consider optimized server distribution policies for predictively compressed light fields. A key difference between that work and our own is the emphasis which we place on distortion-sensitive rendering at the client—something we believe to be completely novel. In the present paper, this also leads us into a novel approach to the representation of geometry through incremental contributions from a disparate set of source depth maps. Our focus on the synthesis of information at the client also helps to decouple the client and server components of the system, as discussed in Section 5.

It is worth mentioning that some answers to the second question posed above have previously been provided for the case in which the entire 3D model, including all texture and geometry components, are to be transmitted over a bandwidth constrained channel. In particular, Tian and AlRegib [18] extend an approach proposed by Balmelli [2], in which the bandwidth assigned to texture and geometry components of a global model are balanced so as to optimize a visualization objective. In these formulations, the global visualization objective either explicitly or implicitly considers a wide range of viewing directions simultaneously. Our approach differs fundamentally from these, in that we are concerned with the optimization of an interactive client's view of interest. Indeed the global perspective becomes increasingly unhelpful as the scene which must be navigated grows. Eventually it becomes unreasonable to expect that any individual client will ever visit more than a fraction of the scene. Moreover, interactive navigation means that the client may move very close to the surface at some instants and much further away at others; these clearly alter the optimal balance between texture and geometry information.

The remainder of the paper is organized as follows. Section 2 develops our proposed distortion-sensitive view synthesis approach. Section 3 extends the approach to include estimates of the local geometric distortion. Section 4 then shows how essentially the same approach may be used to synthesize local surface geometry from a variety of view-dependent depth maps. This approach allows scalable image compression and distribution techniques to be leveraged for both texture and geometry components, treating depth maps as images. Section 5 provides a brief overview of the JPIP interactive communication standard, showing how it may readily be exploited to realize our scene browsing paradigm, while Section 6 considers the complexity of the proposed approach. Finally, Section 7 provides experimental evidence to validate our distortion-based synthesis approach.

## 2. Distortion-sensitive view synthesis

### 2.1. Rendering from a single view

Let $V^*$ denote a desired view. In this section, we briefly discuss the process of rendering $V^*$ from a

single original view image, $V^i$. For the moment, we assume that the surface geometry $G$ is known, having a triangular mesh representation. Later, we will see how to replace the geometry with a synthesized depth map and eliminate the need for a mesh altogether. However, the simplest way to understand things is to begin with a complete mesh.

Let us call $\Delta_n$ the $n$th triangle of the mesh $G$. By projecting the nodes of the mesh onto the image planes corresponding to $V^*$ and $V^i$, we obtain two sets of corresponding triangles, denoted $\{\Delta_n^*\}$ and $\{\Delta_n^i\}$, respectively. This is illustrated in Fig. 2. Isometric or perspective projections might be employed, depending upon our visualization preferences. Of course some of the projected triangles may be hidden in one image, but not the other. If $\Delta_n^*$ is hidden, $\Delta_n^i$ is not involved in rendering, while if $\Delta_n^i$ is hidden, $\Delta_n^*$ is a "hole" in $V^*$, which cannot be rendered from $V^i$. One could consider partially hidden triangles, but this can be avoided by choosing a suitably fine mesh. Equivalently, it is sufficient to subdivide (i.e., remesh) those triangles which straddle object boundaries in $V^*$ or $V^i$. We write $\mathcal{O}^i$ for the set of indices $n$ such that $\Delta_n^i$ is observable in $V^i$—i.e.,

$$\mathcal{O}^i = \{n \mid \Delta_n^i \text{ is observable in } V^i\}; \quad \text{and}$$
$$\mathcal{O}^* = \{n \mid \Delta_n^* \text{ is observable in } V^*\}.$$

Apart from the holes, each exposed triangle $\Delta_n^*$, $n \in \mathcal{O}^*$, is rendered by affine warping of $\Delta_n^i$. We write the overall piecewise affine mapping as $V^* = \mathcal{W}^i(V^i)$. More formally, let $\mathcal{W}_n^i$ be a single affine image warping operator which aligns the imagery in $V^i$ over $\Delta_n^i$ with that in $V^*$ over $\Delta_n^*$. Also, let $V|_\Delta$ denote the image obtained by restricting $V$ to the support of $\Delta$, setting its samples equal to 0 elsewhere. Then the piecewise affine map $\mathcal{W}^i$ is defined by

$$V^* = \mathcal{W}^i(V^i) = \sum_{n \in \mathcal{O}^i \cap \mathcal{O}^*} \mathcal{W}_n^i(V^i)|_{\Delta_n^*}.$$

The interpretation of this is as follows. $\mathcal{W}_n^i$ is a linear operator, involving suitable interpolation kernels, possibly of infinite extent.[2] Although the domain of $\mathcal{W}_n^i$ is notionally restricted to $\Delta_n^i$, the presence of such interpolation kernels requires us to regard $\mathcal{W}_n^i$ as an operation on the entire domain of $V^i$, taking the result only over $\Delta_n^*$. The individual affine warping operators $\mathcal{W}_n^i$ are only defined for those $n$ which belong to both $\mathcal{O}^i$ and $\mathcal{O}^*$. Accordingly, $V^*$ contains two types of "holes," which we identify as exterior and interior holes:

(1) "Exterior holes" consists of those pixels in the support of $V^*$ (typically a rectangular image) which do not fall within the region of support (or silhouette) of the object, as perceived from view $V^*$. We write
$$\mathcal{R}^* = \bigcup_{n \in \mathcal{O}^*} \Delta_n^*$$
for this region of support.
(2) "Interior holes" are defined by the region
$$\mathcal{H}^{i \to *} = \bigcup_{n \in \mathcal{O}^* \setminus \mathcal{O}^i} \Delta_n^*.$$
This is the portion of $\mathcal{R}^*$ which is not visible in view $V^i$.

We conclude this sub-section by remarking that affine warping does not exactly extend the behaviour of a perspective imaging model into the interior of the projected surface triangles $\Delta_n^*$. However, this error can be rendered arbitrarily small by reducing the size of the surface mesh elements. A suitable remeshing is thus sufficient to validate our formulation, with respect to both modelling precision and visibility.

### 2.2. Combining multiple views

One way to combine the information from multiple original view images, $V^{i_0}, V^{i_1}, \ldots$, is to simply average the results obtained by mapping each of them onto the desired view. Unfortunately, any imperfections in the surface geometry

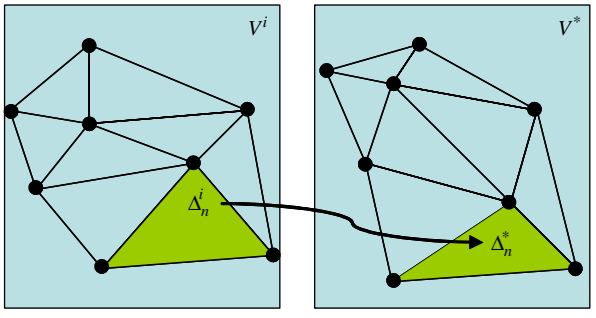


Fig. 2. Surface mesh projected onto $V^*$ and an original view image $V^i$.

[2] As in the case of spline interpolators of quadratic or higher order.

description will produce misalignment amongst the separate renderings $\mathscr{W}^i(V^i)$, so that averaging tends to blur high-frequency spatial features. Also, the simple average shows no preference for one possible rendering over another.

An alternate strategy is to select a single "most appropriate" original view image from which to render each triangle. We refer to this as "stitching," writing $i_n^*$ for the "best stitching source" for the $n$th triangle, and constructing the synthesized view as a patchwork of these best stitching sources, according to

$$V^* = \sum_{n\in(\cup_i \mathscr{O}^i)\cap\mathscr{O}^*} \mathscr{W}^{i_n^*}(V^{i_n^*})|_{\Delta_n^*} = \sum_{n\in\mathscr{O}^*} \mathscr{W}^{i_n^*}(V^{i_n^*})|_{\Delta_n^*}. \quad (1)$$

Of course, $i_n^*$ must have the property that $n \in \mathscr{O}^{i_n^*}$ so that $\Delta_n^{i_n^*}$ is visible in $V^{i_n^*}$. Also, note that the "interior holes" in $V^*$ are now restricted to the portion of $\mathscr{R}^*$ which is not visible from any of the available views $V^i$, i.e.,

$$\mathscr{H}^* = \bigcap_i \mathscr{H}^{i\to *} = \bigcup_{n\in\mathscr{O}^*\setminus(\cup_i\mathscr{O}^i)} \Delta_n^*.$$

An obvious challenge associated with the stitching approach is to identify the best stitching source for each triangle. This is the subject of Sections 2.3 and 3, which consider the impact of both image and geometric modelling distortions on the selection of a suitable candidate.

Although stitching avoids the blurring problem, it tends to produce visible discontinuities at the boundaries between adjacent triangles which are rendered from different original source views. This is because the surface geometry will inevitably contain modelling errors, and the rendering process described here does not account for illuminant-dependent shading effects.

One way to reduce the visibility of stitching boundaries is to perform some averaging in the vicinity of triangle boundaries. This could be done by forming a weighted average of the various candidates, $\mathscr{W}^i(V^i)$, in the vicinity of stitching boundaries. As mentioned above, however, averaging tends to destroy high-frequency spatial features; moreover, it is unclear how much averaging is required to eliminate visible artifacts. A classic solution to this dilemma, which has found wide applicability, is to perform the stitching within a multi-resolution framework [3] such as the Laplacian pyramid or a discrete wavelet transform (DWT). This can be shown to have the effect of providing much more smoothing to lower spatial frequency components than higher-frequency components, thereby strongly concealing transitions while passing higher-frequency content unaltered.[3]

For the present work, we select the DWT for our stitching procedure. This is motivated by the fact that our original view images will be compressed using the DWT-based JPEG2000 standard. The last term in Eq. (1) reveals that stitching may be accomplished by separately creating a complete warped image $\mathscr{W}^i(V^i)$ for each available source image, and then selectively stitching these warped images together. It is convenient to define

$$V^{i\to *} \triangleq \mathscr{W}^i(V^i).$$

For multi-resolution stitching, we first decompose each $V^{i\to *}$ using a $D$ level DWT, forming a low-resolution base image $\mathrm{LL}_D^{i\to *}$ and a collection of high-pass subbands $\mathrm{HL}_d^{i\to *}$, $\mathrm{LH}_d^{i\to *}$ and $\mathrm{HH}_d^{i\to *}$, as shown in Fig. 3. The image $V^{i\to *}$ may be synthesized from its subbands by recursive application of the DWT synthesis operator $\mathscr{S}$, where

$$\mathrm{LL}_d = \mathscr{S}(\mathrm{LL}_{d+1}, \mathrm{HL}_{d+1}, \mathrm{LH}_{d+1}, \mathrm{HH}_{d+1})$$

and $V^{i\to *} \equiv \mathrm{LL}_0^{i\to *}$. This is also depicted in Fig. 3.

Equivalently, noting that $\mathscr{S}$ is a linear operator, we may re-write the recursive synthesis procedure as

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

Here, $\mathscr{S}_{\mathrm{L}}$ and $\mathscr{S}_{\mathrm{H}}$ are the low- and high-pass portions of the overall synthesis operation required to implement a single stage of DWT synthesis, and $R_d$ denotes the "detail" image formed from the three high-pass subbands at decomposition level $d+1$. It is convenient to write $R_D$ for $\mathrm{LL}_D$ so that $R_0^{i\to *}$, $R_1^{i\to *}$, …, $R_D^{i\to *}$ together represent the complete set of resolution components for warped image $V^{i\to *}$. Our multi-resolution stitching algorithm proceeds by separately stitching each resolution component to form

$$R_d^* = \sum_{n\in\mathscr{O}^*} R_d^{i_{n,d}^*\to *}|_{\Delta_{n,d}^*}, \quad d = 0, 1, 2, \ldots, D. \quad (2)$$

Here $\Delta_{n,d}^*$ represents the support of triangle $\Delta_n^*$, as it appears in resolution component $R_d$—i.e., reduced in size by the factor $2^d$. We also note that the "best

[3] In fact, the highest-frequency details in the multi-resolution pyramid are not smoothed at all.

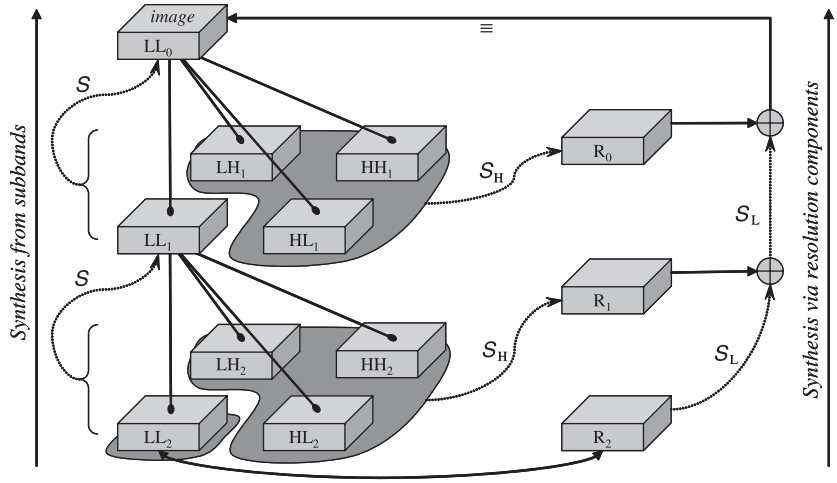


Fig. 3. Relationship between DWT subbands, resolution components and the image which they represent, for the case of a $D = 2$ level decomposition.

stitching source," $i^*_{n,d}$, need not necessarily be the same in every resolution component $R_d$. This is an important property, since one source view might provide high-quality details at some resolutions but not at others, depending upon compression noise and the view orientation. These matters are taken up further in the next sub-section.

Before moving on, we note that the formulation provided in Eq. (2) fails to properly address the presence of "holes" in the various warped views $V^{i\to *}$. The chief difficulty is that the multi-resolution transform involves overlapping basis functions, so that resolution component samples which lie fully within a visible triangle may nevertheless be influenced by the invalid data associated with holes. Rather than complicating matters at this point, we defer the treatment of holes until Section 2.5.

### 2.3. Incorporating distortion information

In this section, we describe a method for selecting the best stitching source $i^*_{n,d}$, for each triangle $\Delta^*_{n,d}$, based solely on the amount of quantization error power which the selection will incur. We thus ignore the limitations of geometric modelling, which are the subject of Section 3. We assume that the original view images have been compressed in the DWT domain (e.g., using JPEG2000). This means that the quantization error in $V^i$ can be expressed as

$$\delta V^i[\mathbf{n}] = \sum_b \sum_{\mathbf{k}} \delta B^i_b[\mathbf{k}] \cdot S^b_{\mathbf{k}}[\mathbf{n}].$$

Here, $B^i_b$ is subband $b$ from image $V^i$, $\delta B^i_b[\mathbf{k}]$ is the error in the $\mathbf{k}$th sample of this subband and $S^b_{\mathbf{k}}$ is the synthesis basis vector (itself an image) for that sample. We use vectors such as $\mathbf{k} = [k_1, k_2]$ and $\mathbf{n} = [n_1, n_2]$ to denote 2D coordinates.

After warping according to $\mathscr{W}^i$ and multi-resolution analysis, we find the quantization error at location $\mathbf{p}$ in resolution component $R^{i\to *}_d$ to be

$$\begin{aligned}\delta R^{i\to *}_d[\mathbf{p}] &= \langle \mathscr{W}^i(\delta V^i), A^d_{\mathbf{p}} \rangle \\ &= \sum_b \sum_{\mathbf{k}} \delta B^i_b[\mathbf{k}] \cdot \langle \mathscr{W}^i(S^b_{\mathbf{k}}), A^d_{\mathbf{p}} \rangle,\end{aligned}$$

where $A^d_{\mathbf{p}}$ is the analysis basis vector (itself an image) associated with that sample, and $\langle \cdot, \cdot \rangle$ signifies the inner product between two images. The total quantization error energy associated with triangle $\Delta^*_{n,d}$ in resolution component $R^{i\to *}_d$ is then

$$\begin{aligned}D^{i\to *}_{n,d} &= \sum_{\mathbf{p}\in\Delta^*_{n,d}} |\delta R^{i\to *}_d[\mathbf{p}]|^2 \\ &\approx \sum_b \underbrace{\sum_{\mathbf{p}\in\Delta^*_{n,d}} \sum_{\mathbf{k}} |\delta B^i_b[\mathbf{k}]|^2 \cdot \langle \mathscr{W}^i(S^b_{\mathbf{k}}), A^d_{\mathbf{p}} \rangle^2}_{D^{i\to *}_{n,b\to d}}, \quad (3)\end{aligned}$$

assuming that the individual subband quantization errors $\delta B^i_b[\mathbf{k}]$ are approximately uncorrelated[4]—a very widespread assumption in the distortion-directed

[4]Uncorrelated quantization errors cause all cross-terms of the form $\sum_{b_1,b_2}\sum_{\mathbf{k}_1,\mathbf{k}_2}\delta B^i_{b_1}[\mathbf{k}_1]\delta B_{b^i_2[\mathbf{k}_2]}$ in the quadratic expression to evaluate to 0.

decision literature. We will henceforth take the above equation as a strict equality.

Now consider the individual contributions $D^{i\to *}_{n,b\to d}$, in the above equation. Due to the decay of the finite support operators in $\langle \mathscr{W}^i(S^b_{\mathbf{k}}), A^d_{\mathbf{p}}\rangle$, $D^{i\to *}_{n,b\to d}$ depends principally on the distortion contributions $\delta B^i_b[\mathbf{k}]$ which are found inside $\Delta^i_{n,b}$, the projection of $\Delta^i_n$ into subband $B^i_b$. With this in mind, we make the simplifying approximation of a uniform quantization error power over the entire subband, equalling the average actual quantization error power $D^i_{n,b}/|\Delta^i_{n,b}|$ over $\Delta^i_{n,b}$. With this uniform error power, we obtain

$$D^{i\to *}_{n,b\to d} = \frac{D^i_{n,b}}{|\Delta^i_{n,b}|}\cdot \sum_{\mathbf{p}\in\Delta^*_{n,d}}\sum_{\mathbf{k}}\langle \mathscr{W}^i(S^b_{\mathbf{k}}), A^d_{\mathbf{p}}\rangle^2 \approx D^i_{n,b}\cdot\frac{|\Delta^*_{n,d}|}{|\Delta^i_{n,b}|}\cdot W^n_{b\to d}. \quad (4)$$

Here, $W^n_{b\to d}$ measures the average value of the expression $\sum_{\mathbf{k}}\langle \mathscr{W}^i{}_n(S^b_{\mathbf{k}}), A^d_{\mathbf{p}}\rangle^2$ over a range of indices $\mathbf{p}$. This is reasonable, since $S^b_{\mathbf{k}}$ and $A^d_{\mathbf{p}}$ are both periodically shift invariant and the single affine operator, $\mathscr{W}^i{}_n$, captures the behaviour of $\mathscr{W}^i$ locally over the triangle in which we are interested. From a practical viewpoint, this enables us to use a pre-computed table of weights, $W^n_{b\to d}$, for each combination of source subband $b$, target resolution component $d$, and affine operator, $\mathscr{W}^i_n$. In practice, we quantize the actual affine parameters to obtain a finite set of indices for our lookup table.[5]

At this point, it is helpful to develop some intuition concerning the expected behaviour of our distortion formulation. Suppose firstly that the DWT synthesis kernels $S^b_{\mathbf{k}}$ are mutually orthonormal.[6] Suppose also that the multi-resolution analysis kernels $A^d_{\mathbf{p}}$ are mutually orthonormal, which must certainly be the case if our multi-resolution transform is derived from the orthonormal DWT, following the procedure outlined in the previous sub-section. Finally, observe that the affine operator $\mathscr{W}^i_n$ serves to stretch each $S^b_{\mathbf{k}}$ by an amount equal to $|\Delta^*_n|/|\Delta^i_n|$, amplifying its energy by "roughly" the same amount (we will revisit this point shortly). From the orthonormality of the $A^d_{\mathbf{p}}$, it follows that

$$\sum_d\sum_{\mathbf{p}}\langle \mathscr{W}^i_n(S^b_{\mathbf{k}}), A^d_{\mathbf{p}}\rangle^2 = \|\mathscr{W}^i_n(S^b_{\mathbf{k}})\|^2 = |\Delta^*_n|/|\Delta^i_n| \quad \forall\mathbf{k}.$$

Now $W^n_{b\to d}$ is the average value of $\sum_{\mathbf{k}}\langle \mathscr{W}^i{}_n(S^b_{\mathbf{k}}), A^d_{\mathbf{p}}\rangle^2$ taken over $\mathbf{p}$. It can be shown that this is $|\Delta^i_{n,b}|/|\Delta^*_{n,d}|$ times the average value of $\sum_{\mathbf{p}}\langle \mathscr{W}^i{}_n(S^b_{\mathbf{k}}), A^d_{\mathbf{p}}\rangle^2$ taken over $\mathbf{k}$, from which could conclude that

$$\sum_d D^{i\to *}_{n,d} = \sum_d\sum_b D^i_{n,b}\cdot\frac{|\Delta^*_{n,d}|}{|\Delta^i_{n,b}|}\cdot W^n_{b\to d} = \sum_b D^i_{n,b}\sum_d\sum_{\mathbf{p}}\langle \mathscr{W}^i{}_n(S^b_{\mathbf{k}}), A^d_{\mathbf{p}}\rangle^2 = \frac{|\Delta^*_n|}{|\Delta^i_n|}\sum_b D^i_{n,b}. \quad (5)$$

At first glance, this would appear to suggest that the total distortion in the warped triangle (left-hand side) should be roughly independent of the affine operator $\mathscr{W}^i_n$, since the total distortion in the source triangle $\sum_b D^i_{n,b}$, should be roughly proportional to its area, $|\Delta^i_n|$. However, two things are missing from this picture, the understanding of which is central to a correct implementation of our distortion-based view synthesis procedure.

The first important oversight in the above derivation is that $\mathscr{W}^i_n$ must be a bandlimited warping operator. While warping a spatially continuous image does indeed amplify its energy directly in proportion to $|\Delta^*_n|/|\Delta^i_n|$, this property can only be preserved for discrete imagery to the extent that all spatial frequency components in the warped image can still be represented. If $|\Delta^*_n|/|\Delta^i_n|<1$, the ideal continuous warping operator necessarily generates extra super-Nyquist spatial frequency components which must be suppressed in the discrete equivalent to avoid aliasing. This means that source views $V^i$ for which $|\Delta^*_n|/|\Delta^i_n|<1$ should yield less distortion power, making them more favourable selections for a single "best stitching source," all other things being equal. This observation also reminds us that care must be taken when warping both the real image samples and the synthesis basis images (for computation of the weights $W^n_{b\to d}$), not to simply resample the image with a fixed interpolation function. One simple way to achieve the desired band-limiting behaviour is to perform our warping operation initially at a higher

[5]Considering that our weights are formed by summing/averaging over all combinations of the locations $\mathbf{p}$ and $\mathbf{k}$, the translation component of $\mathscr{W}^i_n$ is irrelevant. This leaves only four degrees of freedom, which may be expressed in terms of the rotation and expansion of the cardinal axes and then quantized.

[6]The $\frac{9}{7}$ biorthogonal wavelet transform used for our experiments is very nearly orthonormal, subject to appropriate normalization of the subband samples.

resolution and then sub-sample the result, using an appropriate anti-aliasing filter.

The second important oversight in the above derivation arises when $|\Delta_n^*|/|\Delta_n^i|>1$. In this case, $\mathscr{W}_n^i$ is expanding the source view, $V^i$. In this case, the highest-resolution components of $V^*$ cannot be recovered at all, since they depend upon super-Nyquist frequency components in the source view. The absence of these high-frequency components represents a form of extra distortion, in addition to that arising from quantization noise in the compressed source views. One way to capture this effect is to extend the summation on the right-hand side of Eq. (5) to include subbands from a set of hypothetical resolutions above those of the original images, as shown in Fig. 4. This figure also shows how the first problem described above may be addressed at the same time by performing our warping procedure at the higher resolution and then simply discarding the extra resolution components as a form of band-limited downsampling.

The only difficulty with the procedure suggested by Fig. 4 is the estimation of source distortions $D_{n,b}^i$ in the hypothetical subbands $b$. Of course, since these subbands are missing, their distortions must be identical to their energies $E_{n,b}^i$. One way to obtain a conservative estimate for these energies is to project each source image onto the other in turn, taking the maximum of the energy produced by such projections.

Accounting for both of the effects described above, we see that source views for which $|\Delta_n^*|/|\Delta_n^i|$ is smaller, are always preferred over those for which $|\Delta_n^*|/|\Delta_n^i|$ is larger, assuming that the source compression noise power is comparable in all views. This agrees with our intuition, that the source view whose focal plane is most parallel to the scene surface should provide the most information about its texture; this is the view for which the transformation $\mathscr{W}^i$ is most contractive.

Before leaving this section, it is worth noting that the view synthesis procedure developed here is to be performed by the client in a remote browsing application. In order to implement the procedure directly, the client must have access to localized information about the quantization error in each DWT subband of each source image. However, the client does not have access to the original images, with which to compare its compressed version.

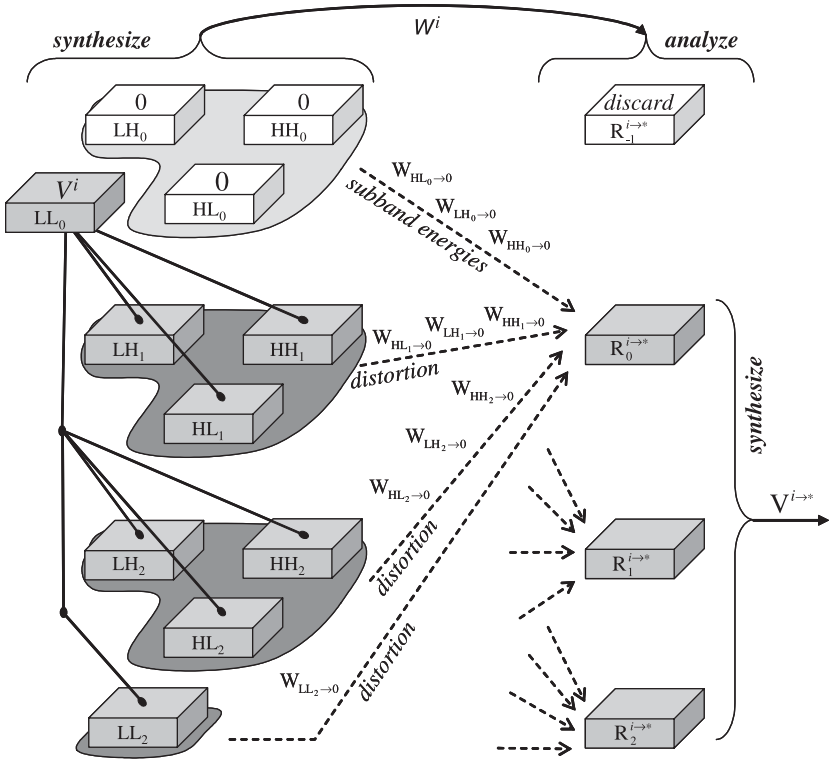


Fig. 4. Procedure for mapping both imagery and distortion information from view $V^i$ to warped view $V^{i\to*}$. Extra, hypothetical resolutions are shown lightly shaded.

It turns out, however, that the client can generally manage a reasonable estimation of the residual local distortion, based only on its compressed representation. This point is considered further in Section 5.

### 2.4. Synthesis from depth maps

Up to this point we have been working with a complete description of the geometry $G$, in terms of triangular mesh elements. From the perspective of view $V^*$, however, all that is actually required is a depth map $Z^*[\mathbf{n}]$, identifying the depth of each location $\mathbf{n}$ in $V^*$. Of course, we can derive such a depth map from the complete mesh. In Section 4, however, we will see how the problem of estimating depth from view $V^*$ can be considered as an analogous problem to that of synthesizing texture from view $V^*$. Shifting our focus to depth maps has a number of benefits. Firstly, it narrows our attention to only that part of $G$ which is relevant to the reconstruction at hand. This is particularly, important to the client–server problem, since we cannot afford to transmit a complete geometry in the case of large, complex scenes. Secondly, by moving to depth maps, we can treat each pixel (or resolution component sample) separately. This circumvents the need for remeshing near boundaries and holes, and allows stitching decisions to follow meaningful scene features rather than artificial mesh structures.

Essentially, all that is required to adjust our previous formulation to the case of depth maps, is to take the limit as triangles $\Delta_n^*$ and $\Delta_n^i$ becomes arbitrarily small. $\mathscr{W}^i$ then becomes a general geometric warping operator, derived from $Z^*$; Eq. (2) translates directly to

$$R_d^*[\mathbf{p}] = \begin{cases} R_d^{i_d^*[\mathbf{p}]\to *}[\mathbf{p}], & 2^d\mathbf{p} \in \mathscr{R}^*, \\ 0, & 2^d\mathbf{p} \notin \mathscr{R}^*, \end{cases} \quad d = 0, 1, 2, \ldots, D; \tag{6}$$

and Eqs. (3) and (4) yield per-sample distortion estimates

$$\begin{aligned} D_d^{i\to *}[\mathbf{p}] &= \sum_b D_{b\to d}^{i\to *}[\mathbf{p}] \\ &= \sum_b W_{b\to d}[\mathbf{p}] \cdot D_b^i[(\mathscr{W}^i_{b\to d})^{-1}(\mathbf{p})]. \end{aligned} \tag{7}$$

This last equation deserves some explanation. First, observe that Eq. (4) may be written as

$$\frac{D_{n,b\to d}^{i\to *}}{|\Delta_{n,d}^*|} \approx \frac{D_{n,b}^i}{|\Delta_{n,b}^i|} \cdot W_{b\to d}^n,$$

meaning that the weights, $W_{b\to d}^n$ map per-sample distortion from source subband $B_b^i$ to warped multi-resolution component $R_d$ over the extent of the $n$th triangle. This is why the weights appear as simple point-wise multipliers in the above formulation for the per-sample distortion contribution $D_{b\to d}^{i\to *}[\mathbf{p}]$. With some abuse of notation, we are using $(\mathscr{W}^i_{b\to d})^{-1}$ for the operator which maps locations $\mathbf{p}$ in the warped resolution component $R_d$, back to the corresponding location $\mathbf{k} = (\mathscr{W}^i_{b\to d})^{-1}(\mathbf{p})$ in subband $B_b^i$ of view $V^i$. The warping operator $\mathscr{W}^i$, along with $(\mathscr{W}^i_{b\to d})^{-1}(\mathbf{p})$ and $W_{b\to d}[\mathbf{p}]$, can all be derived directly from the depth map $Z^*[\mathbf{n}]$ alone.

Following the development in Section 2.3, the best stitching source, $i_d^*[\mathbf{p}]$, at location $\mathbf{p}$ in $R_d^*$, should be selected as the one which minimizes distortion, i.e.,

$$i_d^*[\mathbf{p}] = \operatorname*{argmin}_i D_d^{i\to *}[\mathbf{p}].$$

There are, however, two obvious problems associated with employing Eq. (7) to determine $D_d^{i\to *}[\mathbf{p}]$ directly. The first problem is that $(\mathscr{W}^i_{b\to d})^{-1}(\mathbf{p})$ will generally fall between available distortion samples $D_b^i[\mathbf{k}]$ in subband $B_b^i$, requiring some interpolation. The second is that our weighting formulation was developed based upon the assumption that the triangular mesh elements are large compared with the support of $\langle \mathscr{W}_n^i(S_\mathbf{k}^b), A_\mathbf{p}^d \rangle$—this is certainly not true when we reduce all of the triangular mesh elements to individual samples. The simplest way to address both of these problems is to apply a low-pass smoothing filter to the distortion estimates $D_b^i[\mathbf{k}]$ prior to the application of Eq. (7). This allows us to use nearest neighbour interpolation (i.e., rounding) in connection with $(\mathscr{W}^i_{b\to d})^{-1}(\mathbf{p})$, without any loss of fidelity. Also, the local subband distortion estimates available at the client can at best be made available over regions (code-blocks in the case of JPEG2000 compressed imagery). The distortion field $D_b^i[\mathbf{k}]$ is thus already likely to be varying only slowly, so that further low-pass filtering has little impact, except to reduce the amount of spurious switching between different source views during stitching.

### 2.5. Multi-resolution stitching with holes

As noted previously, the warped source views $V^{i\to *}$, which form the basis for our multi-resolution stitching formulation, are subject to the appearance

of "holes." There are two types of holes, which we have classified earlier as exterior and interior.

Exterior holes are common to all of the warped source images $V^{i\to *}$, since they correspond to pixel locations which do not belong to the silhouette $\mathscr{R}^*$ of the object. Even though all $V^{i\to *}$ are zero outside $\mathscr{R}^*$, this does not mean that the resolution components $R_d^{i\to *}$ are zero outside the corresponding, scaled region. The reason is that the DWT (or any multi-resolution transform for that matter) involves overlapping basis functions, which arise as the translates of a set of analysis filter impulse responses. As a result, synthesizing $V^*$ directly from the resolution components defined by Eq. (6) produces a result whose region of support is not limited to $\mathscr{R}^*$, with ringing near the boundaries of this support. To eliminate this problem, it is sufficient to simply set $R_d^*[\mathbf{p}]$ equal to $R_d^{i_d^*[\mathbf{p}]\to *}[\mathbf{p}]$ for all $\mathbf{p}$ in the domain of $R_d$. We have then only to provide a means for determining a suitable stitching source $i_d^*[\mathbf{p}]$ for each $\mathbf{p}\notin\mathscr{R}^*$. This is mildly problematic because neither $W_{b\to d}[\mathbf{p}]$ nor $(\mathscr{W}^i_{b\to d})^{-1}(\mathbf{p})$ formally exist at these locations. The method adopted here is simply to extrapolate $W_{b\to d}[\mathbf{p}]\cdot(\mathscr{W}^i_{b\to d})^{-1}(\mathbf{p})$, as required. This effectively eliminates the problem of exterior holes, which we shall henceforth ignore.[7]

Interior holes present a very different problem, since they occupy different regions in each of the $V^{i\to *}$. Also, it is not sufficient simply to ensure that the best stitching source $i_d^*[\mathbf{p}]$ corresponds to a view which has no holes at the corresponding location. The reason is again because the multi-resolution transform involves overlapping basis functions. In particular, each sample in $R_d^{i\to *}$ is effectively formed an inner product,

$$R_d^{i\to *}[\mathbf{p}] = \langle V^{i\to *}, A_{\mathbf{p}}^d\rangle,$$

where the analysis kernels $A_{\mathbf{p}}^d$ can readily be found by iterative application of the relevant DWT filters. For the DWT-based transform outlined in Fig. 3, we have

$$A_{\mathbf{p}}^D[\mathbf{n}] = A_{\mathrm{L}}^D[\mathbf{n}-2^D\mathbf{p}], \quad \text{and}$$
$$A_{\mathbf{p}}^d[\mathbf{n}] = A_{\mathrm{H},\mathbf{p}\bmod 2}^d[\mathbf{n}-2^d(\mathbf{p}-\mathbf{p}\bmod 2)], \quad 0\leqslant d<D,$$

where $\mathbf{p}\bmod 2$ is the vector $[p_1 \bmod 2, p_2 \bmod 2]$ and the fundamental low- and high-pass kernels are recursively defined by

$$A_{\mathrm{L}}^0[\mathbf{n}] = \delta[\mathbf{n}] - \text{the unit impulse},$$

$$A_{\mathrm{L}}^{d+1}[\mathbf{n}] = \sum_{\mathbf{k}} h_{\mathrm{L}}[\mathbf{k}]\cdot A_{\mathrm{L}}^d[\mathbf{n}+2^d\mathbf{k}],$$

$$A_{\mathrm{H},\mathbf{p}}^d[\mathbf{n}] = -A_{\mathrm{L}}^d[\mathbf{n}] + \sum_{\mathbf{k}} g_{\mathrm{L}}[2\mathbf{k}+\mathbf{p}]\cdot A_{\mathrm{L}}^{d+1} \times[\mathbf{n}+2^d(2\mathbf{k}+\mathbf{p})], \quad \mathbf{p}\in\{0,1\}^2.$$

Here, $h_{\mathrm{L}}$ and $g_{\mathrm{L}}$ denote the low-pass DWT analysis and synthesis filter impulse responses, respectively. With finite support filters, the dimensions over which $A_{\mathbf{p}}^d$ is non-zero grow roughly as $2^d$. While precise formulation of the region of support $\mathscr{R}_{\mathbf{p}}^d$ for $A_{\mathbf{p}}^d$ is not difficult, the following tight upper bound is convenient for the case of separable symmetric DWT filters, with lengths $2L_{\mathrm{h}}+1$ (analysis) and $2L_{\mathrm{g}}+1$ (synthesis). In our experiments, JPEG2000s $\frac{9}{7}$ DWT is employed, for which $L_{\mathrm{h}}=4$ and $L_{\mathrm{g}}=3$.

$$\mathscr{R}_{\mathbf{p}}^d \subseteq [-l_d,+l_d]^2+\mathbf{p}, \quad \text{with}$$
$$l_d \triangleq \begin{cases}(2^D-1)L_{\mathrm{h}}, & d=D,\\ (2^{d+1}-1)L_{\mathrm{h}}+2^dL_{\mathrm{g}}, & 0\leqslant d<D.\end{cases}$$

Returning now to the problem of interior holes, we note that $R_d^{i\to *}[\mathbf{p}]$ is affected by interior holes in $V^{i\to *}$ whenever $\mathscr{R}_{\mathbf{p}}^d$ intersects $\mathscr{H}^{i\to *}$. Mixing such transform coefficients into the synthesized image will produce results which are generally inconsistent with any of the warped original views. We would like, therefore, to choose the best stitching source $i_d^*[\mathbf{p}]$, from those views $V^i$ such that $\mathscr{R}_{\mathbf{p}}^d$ is fully contained within the complement, $\overline{\mathscr{H}^{i\to *}}$ of $\mathscr{H}^{i\to *}$. This, however, raises two further questions: (1) how should we handle the case where $\mathscr{R}_{\mathbf{p}}^d$ intersects with every $\mathscr{H}^{i\to *}$? and (2) what should be do if the only views $V^i$ for which $\mathscr{R}_{\mathbf{p}}^d\subset\mathscr{H}^{i\to *}$ have an unacceptably high level of distortion (e.g., if they contain no compressed information at all in the region of interest)? Ultimately, we need a way of interpreting overlap between $\mathscr{R}_{\mathbf{p}}^d$ and $\mathscr{H}^{i\to *}$ as an additional source of distortion. At the same time, hard stitching (i.e., selecting only one stitching source for each location $\mathbf{p}$) is likely to produce spurious artifacts in the vicinity of holes, since the stitching sources are not consistent.

[7] There is actually nothing fundamental about exterior holes. If our original geometric description was sufficient to cover the real world, there would be no exterior holes, since $\mathscr{R}^*$ would be infinite.

Our response to the above questions is to extend the hard stitching framework to one in which multiple weighted contributions from different views are permitted in the vicinity of interior holes. Eq. (6) becomes

$$R_d^*[\mathbf{p}] = \sum_i \rho_d^i[\mathbf{p}] \cdot R_d^{i\to *}[\mathbf{p}], \quad d = 0, 1, 2, \ldots, D, \quad (8)$$

where the weights $\rho_d^i[\mathbf{p}]$ sum to 1 at each location $\mathbf{p}$. To preserve the benefits of stitching, we aim to set all but one of the weights to 0 wherever suitable source content is available. To this end, the distortion model of Eq. (7) is first augmented to

$$D_d^{i\to *}[\mathbf{p}] = \sum_b W_{b\to d}[\mathbf{p}] \cdot D_b^i[(\mathscr{W}_{b\to d}^i)^{-1}(\mathbf{p})] + \frac{|\mathscr{R}_{\mathbf{p}}^d \cap \mathscr{H}^{i\to *}|}{|\mathscr{R}_{\mathbf{p}}^d|} \overline{E_d^{i\to *}[\mathbf{p}]}, \quad (9)$$

where $\overline{E_d^{i\to *}[\mathbf{p}]}$ is a local measure of the variance[8] of $R_d^{i\to *}$ in the vicinity of $\mathbf{p}$. This is reasonable, since the presence of holes tends to generate high-energy coefficients in the detail components $R_d^{i\to *}$, and high variance in the low-pass component $R_D^{i\to *}$; these artificial high-energy coefficients are the ultimate source of distortion in the vicinity of holes, since they are inconsistent across views with different hole geometries. Stitching weights are then obtained using

$$\rho_d^i[\mathbf{p}] = \begin{cases} \delta(i - i_d^*[\mathbf{p}]) & \text{if } \mathscr{R}_{\mathbf{p}}^d \cap \mathscr{H}^{i_d^*[\mathbf{p}]\to *} = \emptyset, \\ \dfrac{\frac{1}{\overline{D_d^{i\to *}[\mathbf{p}]}}}{\sum_j \frac{1}{\overline{D_d^{j\to *}[\mathbf{p}]}}} & \text{otherwise,} \end{cases} \quad (10)$$

where $\delta(\,)$ is the Kronecker delta and $i_d^*[\mathbf{p}] = \arg\min_i D_d^{i\to *}[\mathbf{p}]$, as before.

## 3. Accounting for geometric modelling errors

As noted at the end of Section 2.3, if quantization error alone is used to determine the best stitching source, the selected source $V^i$ will tend to be that for which the warping operator $\mathscr{W}^i$ is most contractive. This is the original view whose focal plane is most parallel to the 3D surface at the point in question. While this makes intuitive sense, if the geometric model were highly unreliable we would expect to do better by selecting the original view image which is most closely aligned with the desired view $V^*$; this is the one which for which the rendering process is least dependent on accurate knowledge of the geometry. This reasoning motivates to include geometric modelling errors into our distortion-based rendering formulation. In the ensuring subsections, we identify two aspects of modelling error which are worth capturing.

### 3.1. Depth uncertainty

Uncertainty in the surface geometry translates into uncertainty in the locally affine warping operator. This, in turn, represents a translational uncertainty, which has been studied previously in [15]. Fig. 5 shows schematically how uncertainty in depth $\delta Z_{\mathbf{n}}^*$ can be converted into a corresponding uncertainty in position $\delta\mathbf{n} = \mathbf{n}' - \mathbf{n}$, within the warped image $V^{i\to *}$. In the figure, $\mathbf{x}_{\mathbf{n}}$ denotes the 3D point corresponding to location $\mathbf{n}$ in $V^*$, with depth $Z_{\mathbf{n}}^* = Z^*[\mathbf{n}]$; $\mathbf{x}'_{\mathbf{n}} = \mathbf{x}_{\mathbf{n}} + \delta\mathbf{x}_{\mathbf{n}}$ identifies the true location of the observed point, assuming a depth error of $\delta Z_{\mathbf{n}}^*$. Of course, $\delta\mathbf{x}_{\mathbf{n}}$ and $\delta Z_{\mathbf{n}}^*$ are related through

$$\delta\mathbf{x}_{\mathbf{n}} = \delta Z_{\mathbf{n}}^* / Z_{\mathbf{n}}^* \cdot (\mathbf{x}_{\mathbf{n}} - \mathbf{c}^*),$$

where we are using the notation $\mathbf{c}^*$ and $\mathbf{c}^i$ to denote the focal points associated with views $V^*$ and $V^i$. The positional error $\delta\mathbf{x}_{\mathbf{n}}$ corresponds to a translational shift in view $V^i$ so that the pixel information which should have been mapped to location $\mathbf{n}$ on $V^*$ was actually mapped to location $\mathbf{n}'$, based upon an assumed scene surface which passes through $\mathbf{x}_{\mathbf{n}}$ with normal $\boldsymbol{\nu}_{\mathbf{n}}$. In the figure, point $\mathbf{y}_{\mathbf{n}}$ denotes the corresponding location on this surface; its projection onto $V^*$ yields the location $\mathbf{n}'_{\mathbf{n}}$.

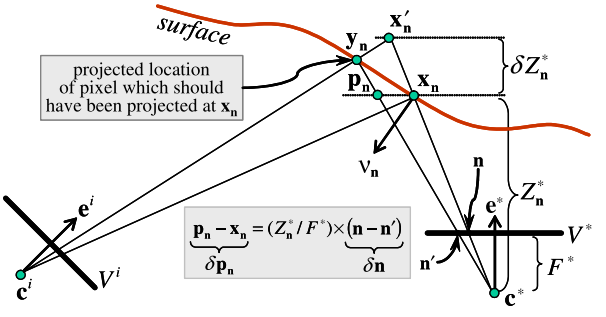


Fig. 5. Relationship between the various coordinates used to map depth uncertainty $\delta Z_{\mathbf{n}}^*$ at location $n$ in $V^*$ into displacement uncertainty $\delta n$ in $V^{i\to *}$.

[8] We say variance here, but for all but the lowest-resolution component ($d = D$), $R_d^{i\to *}$ has essentially zero mean, so we are actually measuring a local average of the power in the coefficients $R_d^{i\to *}[\mathbf{p}]$ for $d < D$.

After some geometry we find that

$$\mathbf{y_n} = \mathbf{c}^i + \frac{\langle \mathbf{x_n} - \mathbf{c}^i, \boldsymbol{\nu}_\mathbf{n} \rangle}{\langle \mathbf{x}'_\mathbf{n} - \mathbf{c}^i, \boldsymbol{\nu}_\mathbf{n} \rangle} (\mathbf{x}'_\mathbf{n} - \mathbf{c}^i),$$

so that for small $\delta Z^*_\mathbf{n}/Z^*_\mathbf{n}$,

$$\begin{aligned}\delta\mathbf{y_n} \triangleq \mathbf{y_n} - \mathbf{x_n} &\simeq \delta\mathbf{x_n} - \frac{\langle \delta\mathbf{x_n}, \boldsymbol{\nu}_\mathbf{n} \rangle}{\langle \mathbf{x_n} - \mathbf{c}^i, \boldsymbol{\nu}_\mathbf{n} \rangle} (\mathbf{x_n} - \mathbf{c}^i) \\ &= \frac{\delta Z^*_\mathbf{n}}{Z^*_\mathbf{n}} \left[ (\mathbf{x_n} - \mathbf{c}^*) - (\mathbf{x_n} - \mathbf{c}^i) \frac{\langle \mathbf{x_n} - \mathbf{c}^*, \boldsymbol{\nu}_\mathbf{n} \rangle}{\langle \mathbf{x_n} - \mathbf{c}^i, \boldsymbol{\nu}_\mathbf{n} \rangle} \right]\end{aligned}$$

and

$$\begin{aligned}\delta\mathbf{p_n} = \mathbf{p_n} - \mathbf{x_n} &\simeq \delta\mathbf{y_n} - \frac{\langle \delta\mathbf{y_n}, \mathbf{e}^* \rangle}{\langle \mathbf{x_n} - \mathbf{c}^*, \mathbf{e}^* \rangle} (\mathbf{x_n} - \mathbf{c}^*) \\ &= \frac{\delta Z^*_\mathbf{n}}{Z^*_\mathbf{n}} \frac{\langle \mathbf{x_n} - \mathbf{c}^*, \boldsymbol{\nu}_\mathbf{n} \rangle}{\langle \mathbf{x_n} - \mathbf{c}^i, \boldsymbol{\nu}_\mathbf{n} \rangle} \\ &\quad \times \left[ (\mathbf{x_n} - \mathbf{c}^*) \cdot \frac{\langle \mathbf{x_n} - \mathbf{c}^i, \mathbf{e}^* \rangle}{\langle \mathbf{x_n} - \mathbf{c}^*, \mathbf{e}^* \rangle} - (\mathbf{x_n} - \mathbf{c}^i) \right],\end{aligned}$$

where $\mathbf{e}^*$ is the view direction for $V^*$ and $\delta\mathbf{n}$ is essentially just $\delta\mathbf{p_n}$ scaled by $F^*/Z^*_\mathbf{n}$, where $F^*$ is the focal length for view $V^*$. Putting all this together, we get

$$|\delta\mathbf{n}|^2 = |\delta Z^*_\mathbf{n}|^2 \cdot \underbrace{\frac{\langle \mathbf{x_n} - \mathbf{c}^*, \boldsymbol{\nu}_\mathbf{n} \rangle^2}{\langle \mathbf{x_n} - \mathbf{c}^i, \boldsymbol{\nu}_\mathbf{n} \rangle^2} \cdot \frac{(F^*)^2 \langle \mathbf{x_n} - \mathbf{c}^i, \mathbf{e}^* \rangle^2}{(Z^*_\mathbf{n})^2 \langle \mathbf{x_n} - \mathbf{c}^*, \mathbf{e}^* \rangle^2} \cdot \left| \frac{\mathbf{x_n} - \mathbf{c}^*}{\langle \mathbf{x_n} - \mathbf{c}^*, \mathbf{e}^* \rangle} - \frac{\mathbf{x_n} - \mathbf{c}^i}{\langle \mathbf{x_n} - \mathbf{c}^i, \mathbf{e}^* \rangle} \right|^2}_{g^{i\to *}_\mathbf{n}}.$$

This shows that variance (error power) in the depth $Z^*_\mathbf{n}$ can be mapped to a corresponding variance (error power) in position, where the factor $g^{i\to *}_\mathbf{n}$ depends upon fixed viewing parameters, together with the surface position $\mathbf{x_n}$ and normal $\boldsymbol{\nu}_\mathbf{n}$ seen at location $\mathbf{n}$ in view $V^*$. We note in passing that $Z^*_\mathbf{n} = \langle \mathbf{x_n} - \mathbf{c}^*, \mathbf{e}^* \rangle$, that the difference between and $\langle \mathbf{x_n} - \mathbf{c}^i, \mathbf{e}^* \rangle$ and $\langle \mathbf{x_n} - \mathbf{c}^*, \mathbf{e}^* \rangle$ is independent of $\mathbf{n}$, and that $\mathbf{x_n}$ can be expressed very simply in terms of $Z^*_n$ and $\mathbf{n}$ by suitable choice of reference coordinates, so that $g^{i\to *}_\mathbf{n}$ is more easily computed than one might at first expect.

Positional uncertainty may further be converted to amplitude distortion, using the method developed in [15]. Specifically, we obtain an additional contribution to the distortion in warped resolution component $R^{i\to *}_d$, of the form

$$|\delta\mathbf{n}|^2 \cdot \frac{1}{(2\pi)^2} \int_{\mathscr{B}_d} \Gamma_{V^{i\to *}}(\boldsymbol{\omega}) \cdot |\boldsymbol{\omega}|^2 \cdot \mathrm{d}\boldsymbol{\omega}, \tag{11}$$

where $\mathscr{B}_d \subset [-\pi, \pi]^2$ is the region occupied by resolution component $R_d$ in the discrete space Fourier domain and $\Gamma_V(\boldsymbol{\omega})$ is the discrete space power density spectrum of image $V$. Of course, the magnitude of $|\delta\mathbf{n}|^2$ varies locally, and we are also interested in local per-sample distortion contributions $D^{i\to *}_d[\mathbf{p}]$. We accommodate this by augmenting Eq. (9) as follows:

$$\begin{aligned}D^{i\to *}_d[\mathbf{p}] = &\sum_b W_{b\to d}[\mathbf{p}] \cdot D^i_b[(\mathscr{W}^i_{b\to d})^{-1}(\mathbf{p})] \\ &+ \frac{|\mathscr{R}^d_\mathbf{p} \cap \mathscr{H}^{i\to *}|}{|\mathscr{R}^d_\mathbf{p}|} \cdot \overline{E^{i\to *}_d[\mathbf{p}]} \\ &+ \overline{|\delta Z^*_d|^2}[\mathbf{p}] \cdot \overline{g^{i\to *}_d}[\mathbf{p}] \\ &\times \overline{|\boldsymbol{\omega}_d|^2} \cdot \overline{E^{i\to *}_d[\mathbf{p}]}.\end{aligned} \tag{12}$$

Here, $\overline{|\delta Z^*_d|^2}[\mathbf{p}]$ is obtained by reducing the resolution of an estimated local distortion field for $Z^*[\mathbf{n}]$ by the factor $2^d$ and applying a suitable low-pass smoothing filter. Similarly, $\overline{g^{i\to *}_d}[\mathbf{p}]$ is obtained by reducing the resolution of the $g^{i\to *}_\mathbf{n}$ field by factor $2^d$ and smoothing the result. All smoothing filters are the same as the one used for $\overline{E^{i\to *}_d[\mathbf{p}]}$, the local average power in $R^{i\to *}_d$ at location $\mathbf{p}$. In Eq. (11), $|\boldsymbol{\omega}|^2$ serves as a frequency-dependent weighting of signal power. In the model of Eq. (12), this is replaced by an assumed "average" weight $\overline{|\boldsymbol{\omega}_d|^2}$, for the whole of resolution component $R_d$. In practice, we use a simple mid-band approximation to get

$$\overline{|\boldsymbol{\omega}_d|^2} = \left( \frac{3\pi}{4} 2^{-d} \right)^2.$$

A variety of other methods to select $\overline{|\boldsymbol{\omega}_d|^2}$ may be found in [15].

Before concluding this sub-section, we note that the true depth map $Z^*$ should be discontinuous at the boundaries of objects which occlude other objects from view $V^*$. Discontinuous depth cannot be accurately represented using discrete samples $Z^*[\mathbf{n}]$ and this problem only becomes worse when depth information is derived from compressed data, as discussed in Section 4. For this reason, it is reasonable to assign a depth uncertainty power, $|\delta Z^*_\mathbf{n}|^2$ which is at least as large as the local variance in $Z^*_\mathbf{n}$. In this way, we implicitly distrust the accuracy of our geometric representation in the vicinity of occluding boundaries. All things (such as

source view distortion) being equal, this encourages the selection of views with smaller values of $g_{\mathbf{n}}^{i\to *}$—these are the views which are closer to $V^*$.

### 3.2. Illuminant uncertainty

Since our surface model does not account for the illuminant-dependent effects of shading and reflection, we can expect a second distortion contribution which grows with the deviation between the orientation of views $V^*$ and $V^i$. Ignoring specularity, we expect this distortion term to be proportional to the signal power, suggesting the following augmented version of Eq. (12).

$$\begin{aligned} D_d^{i\to *}[\mathbf{p}] = \sum_b W_{b\to d}[\mathbf{p}] \cdot D_b^i[(\mathscr{W}^i_{b\to d})^{-1}(\mathbf{p})] \\ + \frac{|\mathscr{R}_{\mathbf{p}}^d \cap \mathscr{H}^{i\to *}|}{|\mathscr{R}_{\mathbf{p}}^d|} \cdot \overline{E_d^{i\to *}[\mathbf{p}]} \\ + \overline{|\delta Z_d^*|^2}[\mathbf{p}] \cdot \overline{g_d^{i\to *}}[\mathbf{p}] \cdot \overline{|\omega_d|^2} \cdot \overline{E_d^{i\to *}[\mathbf{p}]} \\ + \gamma \tan(\max\{0, \cos^{-1}\langle \mathbf{e}^i, \mathbf{e}^*\rangle\}) \cdot \overline{E_d^{i\to *}[\mathbf{p}]}. \end{aligned} \tag{13}$$

Here, $\mathbf{e}^i$ and $\mathbf{e}^*$ are the view directions, as shown in Fig. 5. In the absence of careful modelling, $\gamma$ is a heuristically assigned quantity, which determines the value we place on illumination fidelity. Note that both of the distortion contributions from geometry vary with the local power $\overline{E_d^{i\to *}}$ in the relevant resolution component. The first term varies additionally with depth uncertainty, spatial frequency and properties of the surface normal (through $\overline{g_d^{i\to *}}$), whereas the second term is affected only by the viewing angles $\mathbf{e}^i$ and $\mathbf{e}^*$.

## 4. Distortion-sensitive geometry synthesis

In Section 2, we showed how the client should synthesize view $V^*$ from a collection of available views, based upon local distortion estimates for those views, together with a single depth map $Z^*$ for view $V^*$. In Section 3, we augmented our formulation to accommodate local distortion estimates in this same depth map. Of course $Z^*$ could be compressed at the server as an image and then transmitted progressively to the client, which would estimate its distortion using the same methods it uses to estimate local distortion in the compressed views $V^i$. However, this approach suggests that the server would have to compress a distinct depth map $Z^*$ for each view the client may wish to render. This has all of the same fundamental drawbacks as having the server generate and compress each distinct view $V^*$ which the client may be interested in, which is the problem we have been seeking to avoid. Alternatively, $Z^*$ could be derived from a scalably compressed 3D mesh $G$, which is incrementally transmitted by the server. There exists a substantial body of work on progressive compression of 3D meshes (e.g., [6,14,10]), which could be leveraged to this end.

A conceptually elegant way to address the communication of geometry would be to have the client synthesize the depth map $Z^*$ from a collection of available depth maps, $Z^{i_1}, Z^{i_2}, \ldots$, which the server has already delivered, with varying levels of fidelity and relevance. This approach addresses the problem of communicating originally captured information, since usually 3D scanners deliver data in the form of images with depth information. Those depth maps can always be merged into a unique representation of the world, which can be transmitted in multi-resolution fashion. The approach proposed in this section, however, obviates the need to actually find a complete surface model, which is a hard task if the scene is complex or the content in the database grows as more information from different view points is captured. Furthermore, this approach permits the delivery of only that part of the geometry which is relevant to the required view. This contrasts with the approach proposed in [18], which considers a global geometry and a single surface texture.

The scenario is depicted in Fig. 6, whose similarity to 1 should be immediate. Neither the server nor the client maintain an explicit scene geometry $G$. Instead, they each work with a collection of source images $V^i$ and a collection of depth maps $Z^i$, each compressed as images and communicated incrementally, on demand. The view points associated with the depth maps need not necessarily coincide with the view points associated with the source images. The server must decide how to distribute its available bandwidth between the enhancement of views $V^{i_k}$ which are already partially available at the client, the enhancement of existing depth maps $Z^{i_j}$ which are already partially available at the client, the delivery of a new view, more closely aligned with $V^*$ or the delivery of new depth maps.

At the client, synthesis of depth $Z^*$ from available depth maps $Z^i$ is very closely related to the view synthesis problem. It is a relatively simple matter to

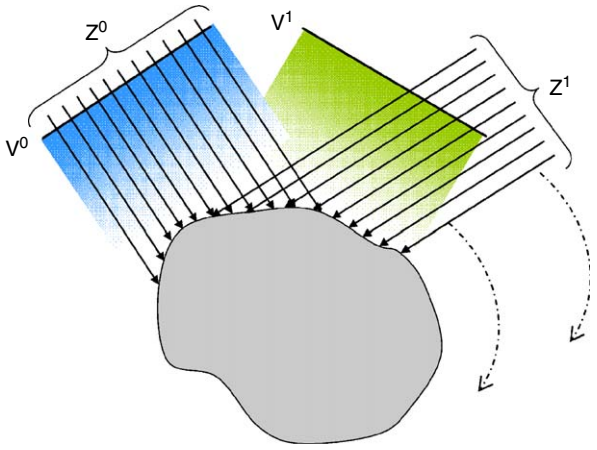


Fig. 6. Browsing environment with a fixed set of views, $V^i$, and depth maps $Z^i$. In this example, $V^0$ and $Z^0$ share the same view point, whereas $V^1$ and $Z^1$ do not.

transform each candidate depth map $Z^i$ into a corresponding estimate $Z^{i\to *}$ for $Z^*$. Our current practical implementation involves building a local triangular mesh for the surface described by $Z^i$ and then warping this mesh into view $V^*$—a task which may be accomplished using the graphic engine on many popular graphics cards. As with view synthesis, expansion or contraction in the local affine warping operators associated with this procedure affect the way in which distortion is mapped from subbands in the compressed $Z^i$ images into the synthesized depth candidate, $Z^{i\to *}$. This is governed by a set of weights, similar to the $W_{b\to d}$ developed in Section 2.3, except that we currently perform depth synthesis directly at full resolution—i.e., without the multi-resolution framework. Local distortion estimates in each $Z^{i\to *}$ are used to guide the stitching procedure for $Z^*$, which is essentially identical to that described in Section 2.3, except that stitching is performed directly in the full resolution pixel domain. The reason for this is that we need to preserve discontinuities in $Z^*$, whereas these tend to be destroyed by multi-resolution stitching. One nice by-product of distortion-sensitive geometry synthesis is that information about local distortion in $Z^*$ is automatically available for inclusion in the view synthesis problem, in accordance with Eq. (13).

It is worth considering explicitly how "holes" manifest themselves in the depth synthesis problem. Each individual depth map $Z^i$ can generally be expected to exhibit strong discontinuities in regions of object occlusion and indeed these discontinuities do correspond to interior holes in the inferred depth

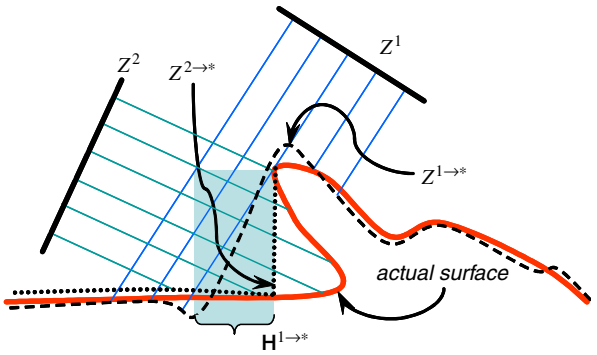


Fig. 7. Illustration of the effect of discontinuities in individual depth maps $Z^i$ on the synthesized depth candidates $Z^{i\to *}$. In the figure, the depth of interest $Z^*$ which we wish to synthesize is the vertical height of the surface.

map $Z^{i\to *}$. Compression using waveform coders such as JPEG2000 and JPEG, however, tends to produce ringing and considerable error in the vicinity of such discontinuities. This phenomenon is illustrated schematically in Fig. 7. In the figure, compressed depth maps $Z^1$ and $Z^2$ are each used to construct candidates $Z^{1\to *}$ and $Z^{2\to *}$ for $Z^*$, where $Z^*$ corresponds to the vertical elevation of the scene surface in this example. Evidently, $Z^1$ should have a discontinuity at the point where the scene surface folds back upon itself, but this discontinuity is corrupted by ringing and loss of high-frequency details due to compression. As a result, it is not possible to detect the hole $\mathscr{H}^{1\to *}$ which should appear in $Z^{1\to *}$. In fact, holes in $Z^{i\to *}$ are difficult if not impossible to detect, based on $Z^i$ alone. The missing information is completed by a second map $Z^2$, for which the inferred depth $Z^{2\to *}$ is shown as a dotted line in the figure.

Fortunately, our distortion-based stitching procedure should recognize that $Z^{1\to *}$ is subject to a great deal of distortion due to the stretching which $Z^1$ undergoes in the vicinity of surface folding. To encourage this, we ensure that the distortion estimates used for each source depth map $Z^i$ have the property that distortion is judged to be at least as large as the local variance of that depth map, in addition to any estimates of the underlying compression noise. This is the same rule which we outlined at the end of Section 3.1 for $Z^*$. The rule serves to ensure that regions in which depth map $Z^1$ was originally discontinuous will be treated with distrust, which is then further amplified by the stretching as $Z^1$ is mapped to $Z^{1\to *}$, encouraging the selection of the much more reliable candidate

$Z^{2\to *}$ in such regions. The final stitched depth map can have much less uncertainty than either of the candidates $Z^{i\to *}$ in isolation, allowing for high-quality view synthesis via the methods of Section 2.

## 5. Client–server communications and distortion estimation

Our envisaged client–server paradigm consists of a collection of original source views $V^i$ and original depth maps $Z^i$, each compressed using JPEG2000 [7] and served using the JPIP standard [9] for remote browsing of JPEG2000 images. JPIP is well suited to the needs of our scene browsing application, but this is unlikely to be obvious to the reader. For this reason, we devote some effort here to a brief review of how JPIP works. Most importantly, we emphasize the freedom which JPIP offers the server to select its own transmission schedule. For a general discussion of JPIP, see [17].

### 5.1. JPEG2000 elements and JPIP data-bins

JPEG2000 is a highly scalable image compression standard, meaning that the compressed representation contains numerous embedded subsets, each of which is an efficient representation of the original image at some reduced resolution, reduced quality, or over a reduced spatial region of interest. The basic elements of this highly embedded representation are illustrated in Fig. 8. Each image is subjected to a DWT, whose subbands are partitioned into "code-blocks," typically measuring $32 \times 32$ samples each for JPIP applications. Each code-block is itself subjected to an efficient fractional bit-plane coding procedure which produces a distinct finely embedded bit-stream. The embedded bit-streams may be truncated at any desired point, allowing a trade-off between compression distortion and coded length; moreover, this truncation may occur at any point after compression.

Code-blocks are organized into spatially coherent regions in each resolution component, known as "precincts." The code-block bit-streams associated with each precinct are then formed into JPEG2000 packets. Each packet contains incremental contributions (possibly empty) from each code-block in the corresponding precinct, so that the set of all packets for a precinct can eventually represent the original code-block samples losslessly. Absence of one or more trailing packets is equivalent to truncation of the original code-block bit-streams; JPEG2000 content creators usually arrange for this truncation to be rate-distortion optimal, in the sense that the best possible image quality (lowest distortion) is obtained for a given overall bit budget, by discarding the same number of trailing packets from all precincts.

JPIP does not deal with JPEG2000 code-blocks or packets directly. Instead, all of the packets

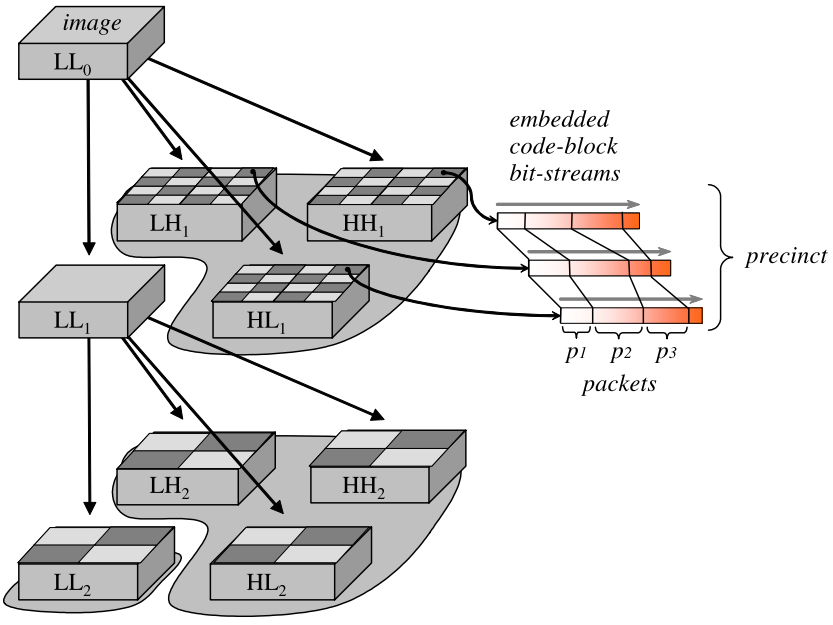


Fig. 8. Basic JPEG2000 elements communicated by JPIP.

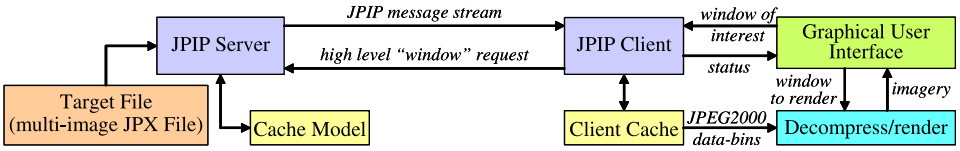


Fig. 9. Client–server interaction in JPIP.

associated with a precinct are first concatenated to form a single "precinct data-bins." Each precinct data-bin thus represents information from the image which contributes to a limited spatial region within a specific resolution component, with the property that the associated image quality can be progressively increased by sending more bytes from the data-bin.

JPIP also defines other types of data-bins, for encapsulating compressed data headers and metadata. This allows JPIP to be used to communicate rich content, in a progressive and selective manner. Of great interest to us is the JPX file format, defined in JPEG2000 Part 2 [8]. Amongst other things, this format allows a single file to encapsulate any number of JPEG2000 compressed images, with a rich and customizable metadata structure. JPIP can represent any JPX file in terms of a large collection of data-bins, with the property that only a small subset of these data-bins are actually required to interpret individual images within the file. This allows us to represent a full set of source views and depth maps within a single JPX file, which can then be served dynamically to a client. The Kakadu software tools[9] provide a comprehensive set of services for creating and remotely interacting with such JPX files.

### 5.2. Client–server interaction in JPIP

Fig. 9 provides a framework for understanding the interaction between a JPIP server and its client. Importantly, the client does not explicitly request data-bins from the server. Instead, the server essentially streams JPIP messages to the client, where each JPIP message consists of a single byte range from a single data-bin. This allows the server to augment the client's local cached representation of the source material. The client may render the content at any time, based on its current cache contents. Importantly, JPEG2000 content can be rendered from any arbitrary subset of the precinct data-bins which might be available, so that rendering at the client can be completely asynchronous with server communications. Each time new data arrives from the server, the client may attempt to render a higher-quality result, updating an interactive user's display.

Although the client does not have direct control over the messages which are streamed from the server, it does nevertheless issue requests. JPIP defines a request language which allows clients to express what they are interested in, using comparatively high-level descriptors. The server then attempts to satisfy the client's interests in the most efficient way, by streaming appropriate messages to augment the client's cached data-bin contents. To facilitate efficient communication, JPIP servers typically maintain a model of the client's cache, so that only relevant increments are actually sent.[10] As the client's needs change, the server adjusts its streaming policy, but exactly when and how it makes these adjustments is left to the server to decide.

For the 3D scene browsing application at hand, the JPIP communication paradigm is a particularly good fit. It allows the server to form its own decisions regarding the best way to improve a synthesized view at the client, either by augmenting the quality of existing views (or depth maps) for which data is already available at the client, or by sending information from new views (or depth maps), for which nothing is currently available in the client's cache. Although we do not specifically study server optimization policies in this paper, it is clear that sufficient flexibility exists to implement a wide range of policies. What is important is that the client's procedures for view synthesis and geometry synthesis eventually utilize information which the server chooses to communicate in response to a requested view $V^*$. Our distortion-based synthesis procedures ensure that this should happen for most reasonable server policies, since view and/or depth

[9] See ⟨http://www.kakadusoftware.com⟩.

[10] JPIP also defines communication modalities which are not session-oriented, along with mechanisms for clients to efficiently signal their cache contents back to the server, but these modes are not relevant to the present paper.

information which need not be heavily warped during synthesis will eventually be used by the client once the associated distortion becomes sufficiently small.

It is worth noting that JPIP does not currently provide a request syntax which can be used to explicitly identify the view $V^*$ of interest for 3D browsing. However, this is something which could quite easily be adopted into the standard as an extension, since high-level requests are already the norm for JPIP.

### 5.3. Distortion estimation vs. explicit signalling

Up until now, we have assumed that local distortion information would be available to the client, for each subband in each view $V^i$ and each depth map $Z^i$. Of course, this distortion is progressively reduced as JPIP messages are delivered by the server. One method for facilitating the generation of distortion estimates $D_b^i[\mathbf{k}]$ at the client would be to arrange for each JPIP message to carry additional distortion information. In particular, noting that each JPIP message communicates a range of bytes for a given data-bin, the message could explicitly signal the amount of distortion incurred when the corresponding precinct is decoded using all information up to that communicated by the message. Since JPIP precinct data-bins typically provide information for only three code-blocks (see Fig. 8), each message could conceivable carry information about the distortion associated with each of these blocks. An efficient strategy for communicating this information approximately might add only a few bytes to the length of each message, although this would require the definition of new message types as an extension to the existing JPIP standard.

As an alternative to explicit communication of the distortion information, it is possible for the client to directly estimate the distortion in each code-block, based solely upon the available data-bin contents. Methods for residual distortion estimation in JPEG2000 code-blocks are studied much more carefully in [16]. The conclusion from that work is that it is possible to estimate the residual MSE distortion in individual code-blocks, typically to within about a factor of 2 (i.e., 3 dB). This may well be sufficient for effective view synthesis. For present experimental work, however, we supply our distortion-based synthesis procedures with actual measured distortion values.

Before concluding this section, we note that neither of the above methods is able to estimate local distortion in subbands for which nothing has yet been communicated. In this case, of course, $D_b^i[\mathbf{k}]$ can be replaced by $E_b^i[\mathbf{k}]$, the estimated energy around location $\mathbf{k}$ in subband $B_b$ of view $V^i$. These energies are also needed to compute the quantities $\overline{E_d^{i\to *}[\mathbf{p}]}$ in Eq. (13), in the absence of any data from the relevant source subband samples. One way to obtain a conservative estimate for these energies, is to project each source image onto the other in turn, taking the maximum of the energy produced by such projections. This is the same procedure outlined at the end of Section 2.3 for estimating the energies associated with the hypothetical super-Nyquist subbands in Fig. 4.

## 6. Complexity considerations

The client rendering procedure proposed in this paper essentially consists of three stages. In the first stage, all available views $V^i$ are decompressed, reprojected onto the target view using the available surface geometry, and decomposed into resolution components $R_d^{i\to *}$. To understand the true complexity of this process, we should first recognize that only that portion of $V^i$ which is visible in $V^*$ needs to be decompressed and reprojected; for many source views, this portion might be empty. We should also recognize that there is no need to decompress $V^i$ at full resolution if the projection operator which maps $V^i$ to $V^*$ is strongly contractive. Conversely, if the projection operator is strongly expansive, there is no need to generate the highest-resolution components $R_d^{i\to *}$, since these should be close to 0; in this case, decompression and warping can all be performed at reduced resolution. Since the reconstruction of JPEG2000 images can be efficiently limited to a given region and resolution of interest, we conclude that the complexity of this first stage is roughly proportional to

$$\sum_i \min\{\|\mathscr{R}^{i\to *}\|, \|\mathscr{R}^{*\to i}\|\}, \tag{14}$$

where $\|\mathscr{R}^{i\to *}\|$ denotes the size (number of pixels) in the portion of $V^*$ which is visible from $V^i$ and $\|\mathscr{R}^{*\to i}\|$ denotes the number of pixels in $V^i$ which are visible from $V^*$. The first term applies when the reprojection operator is strongly contractive, while the second term applies for strongly expansive reprojection, in which $V^{i\to *}$ is synthesized only at reduced resolution.

Admittedly, the complexity expression provided above does not account for reprojections which contain both strongly expansive and strongly contractive elements. However, it does tell us that the complexity associated with synthesizing $V^*$ from a large number of very close (and hence narrowly focused) source views is essentially the same as the complexity of synthesizing $V^*$ from a small number of far away (and hence widely spread) source views, at least in the first stage. This means that the complexity depends only on the size of the reconstructed view $V^*$ and the degree of overlap (or redundancy) between the relevant source views. In our present experimental implementation, we do not exploit the opportunity to perform reconstructions at reduced resolution, on which the above expression is based. Also, our current implementation is in no way optimized, so that this first step currently consumes 2.7 s to reproject four source views at a resolution of $1024 \times 768$ onto a window which also measures $1024 \times 768$, and 1.9 s on a window of $512 \times 384$. Only four source views are considered, since other views, being behind the object, are not involved in reprojection.

In the second stage, the distortion and energy fields associated with each source view subband are projected into the target resolution components using the weighting procedure illustrated in Fig. 4. It is not hard to see that the complexity of this process also follows the expression in (14). To see this, observe that under strongly contractive mappings there is no need to include the contribution from high-frequency source view subbands, while under strongly expansive mappings no source view subband makes a significant contribution to the highest target resolution components. Again, therefore, the overall complexity is dominated by the size of the reconstructed view $V^*$ and the degree of overlap between relevant source views, regardless of the actual number of source views or their original sizes. Additionally, the smooth nature of the available subband distortion fields suggests that this stage in the process could be performed on a sparse grid. Our current implementation, however, does not exploit these various opportunities for complexity minimization.

In the final stage, blending weights are computed from the projected distortion information and the blended resolution components are synthesized to form $V^*$. The complexity of this process would appear to be proportional to the product of the number of pixels in $V^*$ and the number of views being blended. However, we can again appeal to the fact that strongly expansive reprojection operators yield essentially no contribution to the higher-resolution components $R_d^{i\rightarrow *}$ so that these contributions need not be considered during blending. Exploiting this fact requires sophisticated data management structures, which we have not currently implemented. Reprojection of distortion and energy fields and computation of blending weights currently consume 10.1 s to reproject four source views at a resolution of $1024 \times 768$ on a window of $1024 \times 768$, and 2.5 s on a window of $512 \times 384$, being proportional to the number of samples in the rendered view. Synthesis of $V^*$ consumes 0.8 s with four source views at a resolution of $1024 \times 768$ on a window $1024 \times 768$, and 0.2 s on a window of $512 \times 384$. The analysis provided above, however, suggests that a careful implementation could potentially operate at multiple frames per second, which would be quite sufficient for interactive browsing.

In the preceding discussion, we have assumed that surface geometry is already available. If the geometry is communicated via depth maps $Z^i$ rather than a global surface mesh, as advocated in Section 4, a precursor stage is required to reconstruct $Z^*$. The complexity considerations for this process are essentially identical to those associated with reconstructing $V^*$ from the $V^i$.

## 7. View and geometry synthesis experiments

We use both synthetic and real 3D models for our rendering experiments. The models in Fig. 10 (Santa Claus) and Fig. 11 (Frog) are obtained with a passive modelling method [1]. Passive modelling procedures lead to geometric error where texture is missing or illumination causes artifacts (such as shadows or specular reflections). That error is taken into account in Eq. (13). We also used synthetic models of Goku and Car, shown in Fig. 12. In our experiments, we have used real pictures of the real models and rendered images of the synthetic models.

First of all, we show the benefits of the DWT stitching procedure described in Section 2.2. When reprojecting images over the geometry, lack of consistency between images $V^i$ which are projected on adjacent patches leads to artifacts. Evidence of the advantages of multi-resolution stitching can be seen in Fig. 13. In (13)(a) stitching is performed in the image domain and edges between different

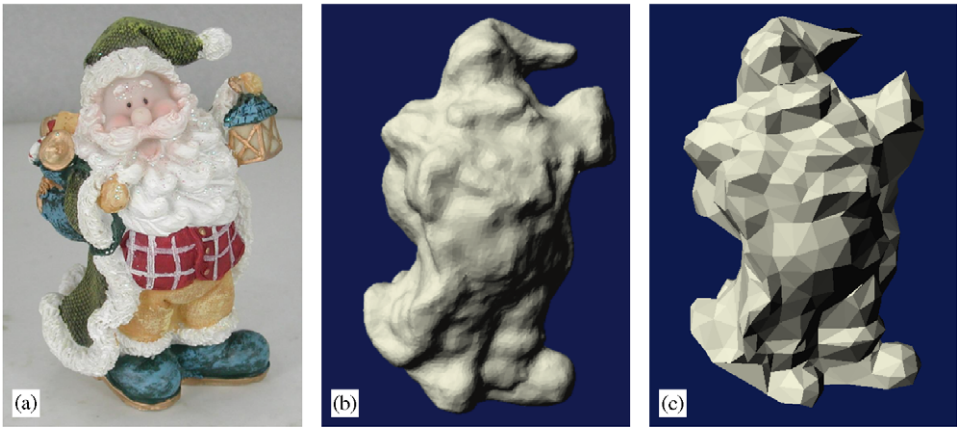


Fig. 10. Santa Claus: (a) one view; (b) the 3D model; and (c) simplified 3D model.

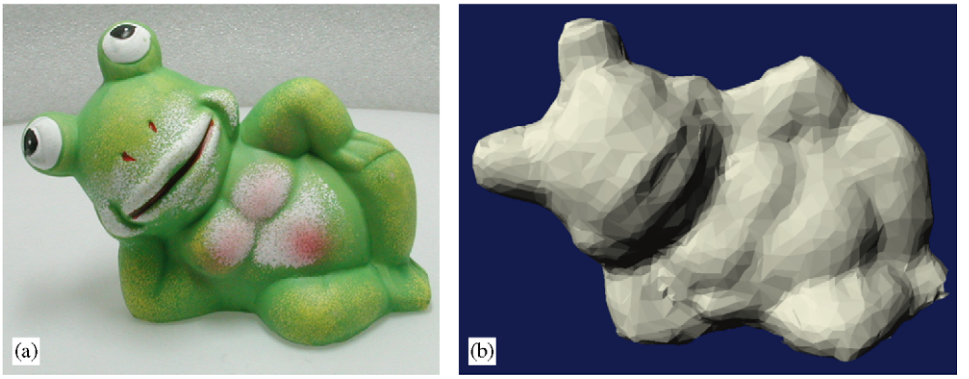


Fig. 11. Frog: (a) one view; and (b) the 3D model.


Fig. 12. Synthetic models used for experiments: (a) Goku; and (b) car.

patches are evident. Edges are mainly due to differences of illumination and geometric mismatch in reprojection. In 13(b) averaging is performed, details are blurred and high frequencies are lost. In 13(c) multi-resolution stitching is employed and artifacts disappear.

The 3D model of Santa Claus is simplified to 1000 triangles, as shown in Fig. 10(c). A simplified model allows the stitching decisions to be more clearly evidenced. At the client side, four images are available, $V^0, \ldots, V^3$, corresponding to views separated by 90°. We initially compress all images to the same high bit-rate of 0.8 bits/pixel (bpp). Our objective is to render $V^*$ from the same view point shown in Fig. 14(a); this view point does not correspond to any of the images available at the client. Fig. 14(b) shows the result of stitching in the image domain, which leads to severe artifacts. Fig. 14(d) shows the results obtained by multi-resolution stitching, following Eqs. (8)–(10). Fig. 14(c) shows the triangles selected by this policy,[11] with different colours for each source view $V^i$.

[11]For illustrative purposes, we select a single best stitching source for each triangle $\Delta_n^*$, over all resolutions. The more general

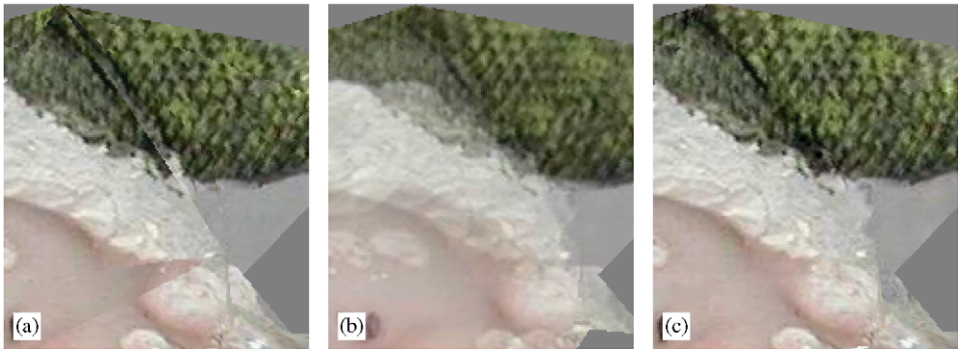


Fig. 13. (a) Image domain stitching; (b) averaging; and (c) multi-resolution stitching.

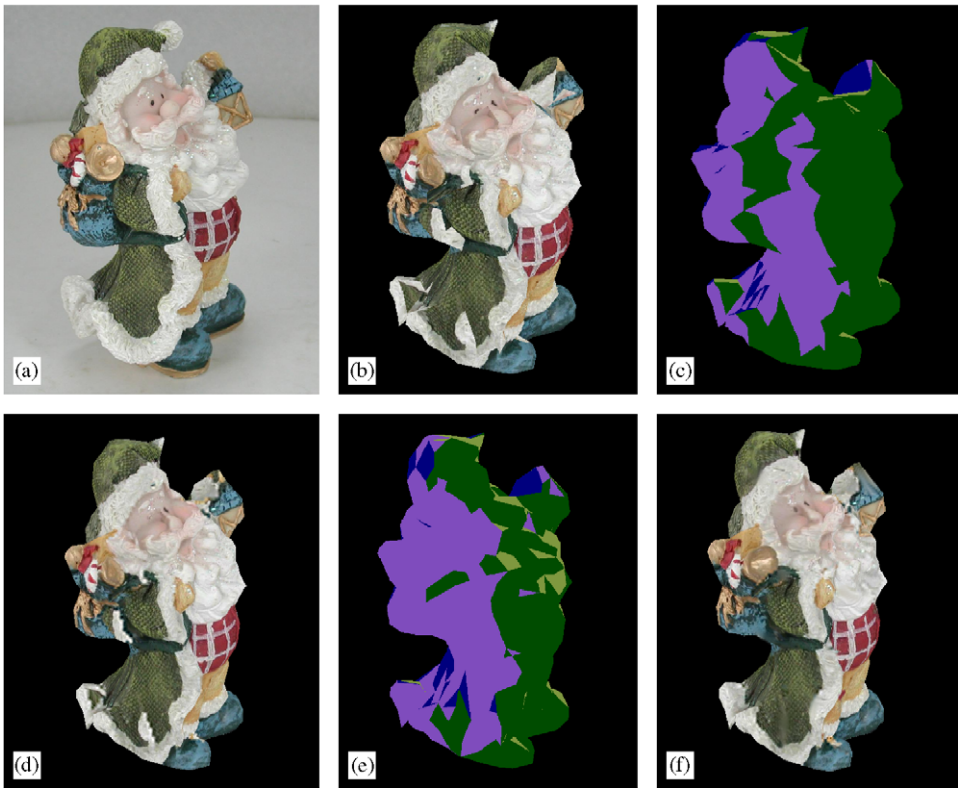


Fig. 14. Santa Claus: (a) reference image; (b) rendering with image domain stitching; (c) and (d) triangle choice and rendering resulting from Eq. (9); (e) and (f) triangle choice and rendering resulting from Eq. (13).

Multi-resolution stitching avoids certain artifacts; however the contribution of geometry modelling errors should also be taken into account. The full distortion formulation of Eq. (13) leads to the reconstruction shown in Fig. 14(f), with the corresponding triangle choice depicted in Fig. 14(e). Geometric uncertainty, causes some triangles to be taken from a view more parallel to the viewer (the purple one in Fig. 14(e)); this is principally due to the influence of the last term in Eq. (12). Improvements in image quality may readily be seen through the disappearance of artifacts at the bottom of the cloak; the face is also rendered more accurately.

(*footnote continued*)
formulation allows for different decisions to be made in each resolution, but this would be very difficult to illustrate here.

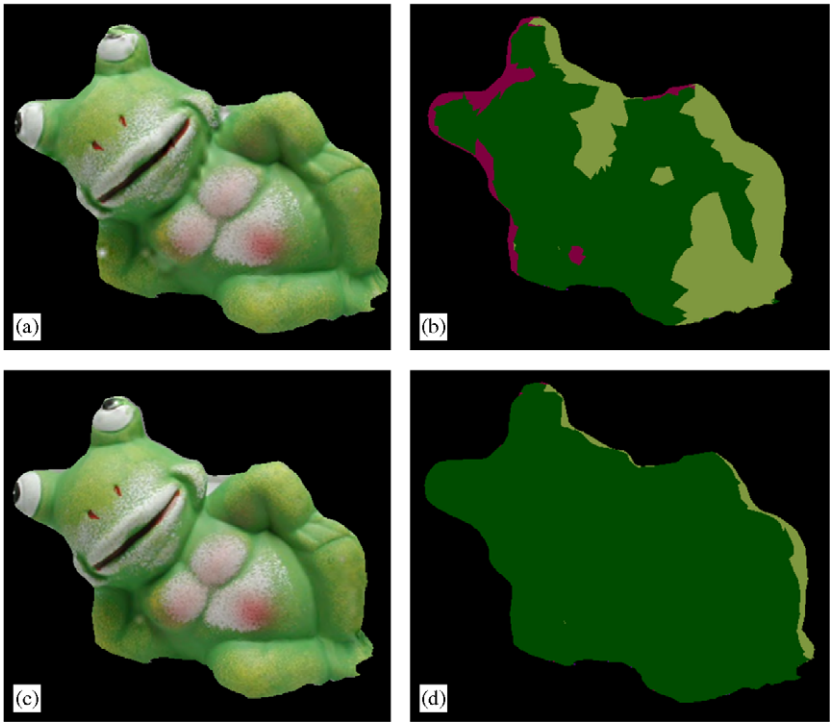


Fig. 15. Frog: (a) and (b) rendering and triangle choice resulting from Eq. (9); (c) and (d) rendering and triangle choice resulting from Eq. (13).

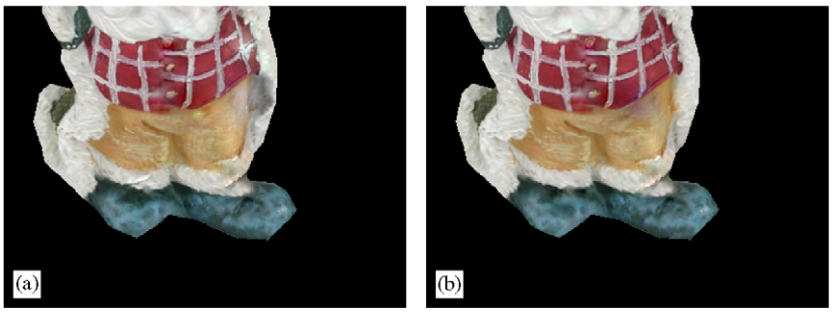


Fig. 16. Artifacts due to internal holes and their removal.

Another example is shown in Fig. 15, this time with the Frog model. Four images are available at the same bit-rate, where one of them is very close to the desired view. Stitching based on quantization distortion effects alone—Eq. (9)—leads to the choice of triangles shown in Fig. 15(b) and to the rendering of Fig. 15(a). Since the frog model has quite a high geometric error, stitching based on the full distortion formulation of (13) leads to the triangle selection shown in Fig. 15(d) and the rendering of Fig. 15(c). When geometry is assigned a high contribution, most of triangles are taken from the view which is more parallel to the viewer. This is evident in Fig. 15(d) where most of the triangles come from the source marked as dark green. In Fig. 15(c), artifacts due to the wrong geometry, visible on the neck of Frog of Fig. 15(a), are no longer present.

Another source of distortion is interior holes in warped images, as discussed in Section 2.5. For example the gray spot on the right in Fig. 16(a) is caused by some low-resolution samples which lie close to interior holes; these samples have large region of support $\mathscr{R}_{\mathbf{p}}^{d}$. By introducing the final term in Eq. (9), the spot disappears from the rendered image, as seen in Fig. 16(b).

If images are sent in a progressive manner, usually they are not available at the client side

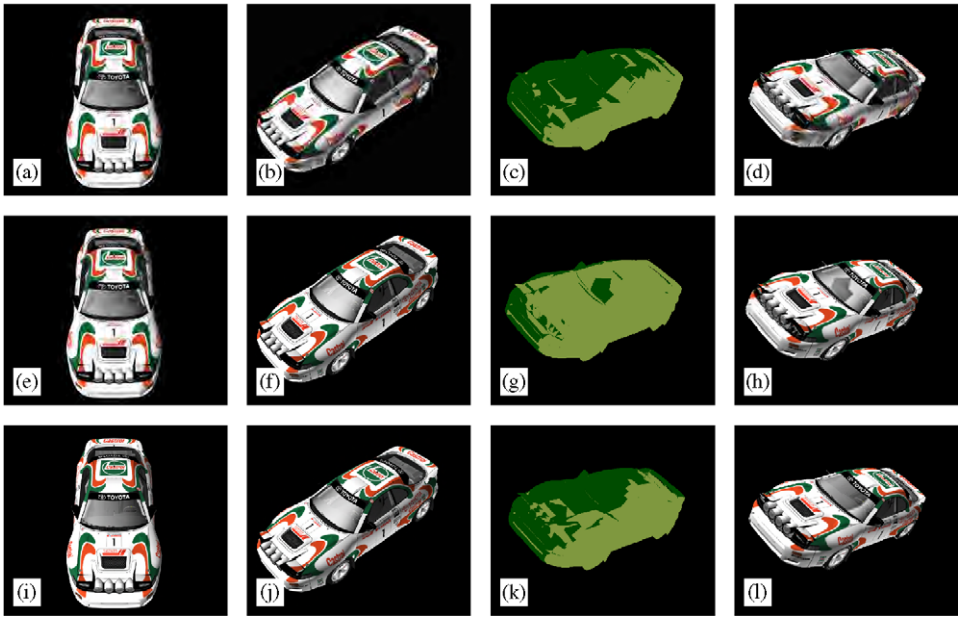


Fig. 17. GT: (a) and (b) $V^{1\prime}$ and $V^{2\prime}$, each at 0.025 bpp; (c) and (d) stitching decisions and rendered view based on $V^{1\prime}$ and $V^{2\prime}$; (e) and (f) $V^{1\prime}$ and $V^{2\prime\prime}$ at 0.025 and 0.4 bpp, respectively; (g) and (h) stitching decisions and rendered view based on $V^{1\prime}$ and $V^{2\prime\prime}$; (i) and (l) $V^{1\prime\prime}$ and $V^{2\prime\prime}$, each at 0.4 bpp; (m) stitching decisions and rendered view based on $V^{1\prime\prime}$ and $V^{2\prime\prime}$.

with equal quality. Images $V^{1\prime}$and $V^{2\prime}$ shown in Fig. 17(a) and (b) are sent to the client at low bit-rate (0.025 bpp). Fig. 17(c) shows the stitching decision and Fig. 17(d) shows the final rendered image $V^*$. Triangles are distributed roughly equally between $V^{1\prime}$and $V^{2\prime}$ since the image distortions are similar. Then, the server sends enhancement information for $V^2$ resulting in the high-quality source view $V^{2\prime\prime}$ (0.4 bpp) shown in Fig. 17(b). As can be seen in 17(g), more triangles are taken from $V^{2\prime\prime}$, since it leads to lower overall distortion in the rendered result. The resulting reconstruction, shown in Fig. 17(h), exhibits significantly improved quality, since most of the low-quality source view $V^{1\prime}$ is discarded. Finally, the server sends enhancements for $V^1$, leaving the client with a second high-quality (0.4 bpp) source view, $V^{2\prime\prime}$, shown in Fig. 17(i). The new stitching decisions are depicted in Fig. 17(m), leading to the rendered view shown in Fig. 17(n), with high-quality details on the entire surface.

In the preceding experiments, distortion has been computed on the basis of individual triangles from an overall geometry, rather than depth maps. As mentioned in Section 4, depth maps provide a much more flexible solution for the interactive viewing of 3D scenes, since both the geometry and the texture information can be augmented at any time by adding more depth maps and/or more source views to the server's data base. In this case, however, our distortion-based synthesis paradigm must be extended to the synthesis of geometry. For the ensuing experiments, we use synthetic models, so that a consistent set of view images $V^i$, and depth maps $Z^i$, can be generated. For each model, we computed ten pairs of view and depth images: eight of them are spaced 45° apart on a circle around the object, one is taken below and one above. All $Z^i$ and $V^i$ are compressed using JPEG2000 and stored in a single JPX file. Some depth maps $Z^i$ and images $V^i$ are shown in Fig. 18, at a variety of quality levels.

When rendering takes place, a new depth map $Z^*$ must be synthesized from the existing ones $Z^1$, $Z^2, \ldots$ . The quality of these available depth maps affects the quality of the reconstructed depth map $Z^*$, as shown in Fig. 19 with different bit-rates for the available $Z^1, Z^2, \ldots$ . Since $Z^*$ takes contributions from multiple depth maps, its quality is generally higher than that of any of the individual source depth maps; this can be seen by comparing 18(b) with 19(c), each of which correspond to a bit-rate of 0.025 bpp.

When our distortion-based stitching procedure generates $Z^*$, the distortion in every depth map is taken into account. High-distortion results from low-quality compression or local expansion in the mapping from $Z^i$ to $Z^*$, or a combination of both.

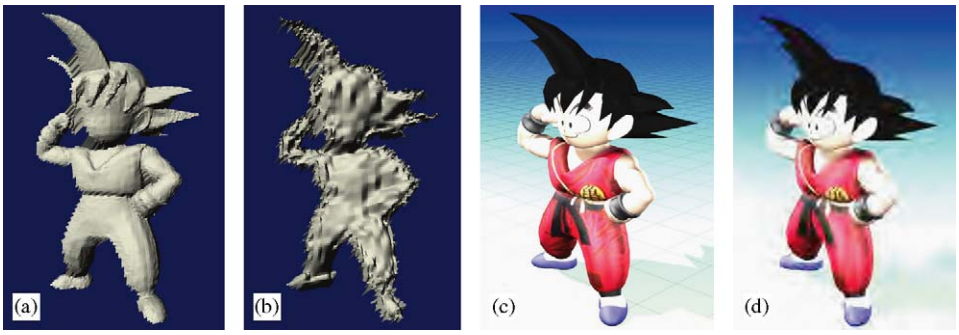


Fig. 18. Some of the Goku view and depth images available at different quality levels: (a) depth map $Z^1$ at 0.4 bpp; (b) depth map $Z^1$ at 0.025 bpp; (c) view $V^2$ at 0.4 bpp; and (d) image $V^2$ at 0.025 bpp.

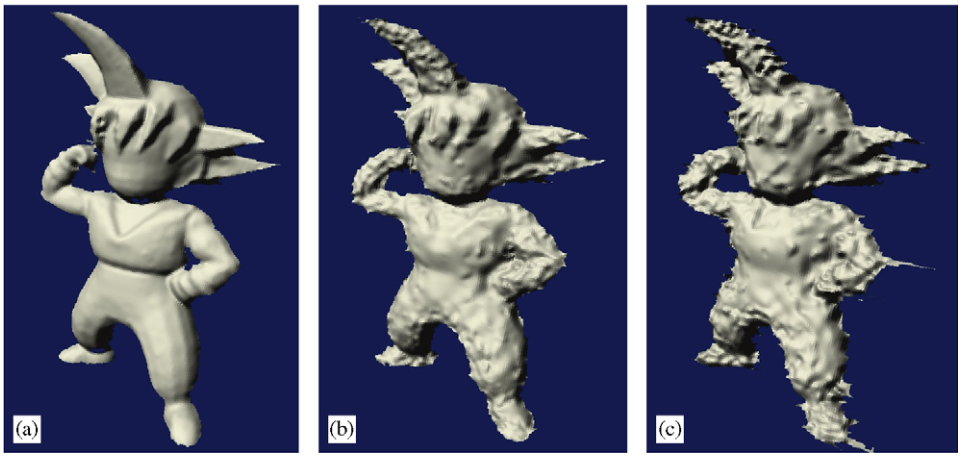


Fig. 19. Synthesized depth map $Z^*$: (a) from 0.4 bpp source maps; (b) from 0.05 bpp source maps; and (c) from 0.025 bpp depth maps.

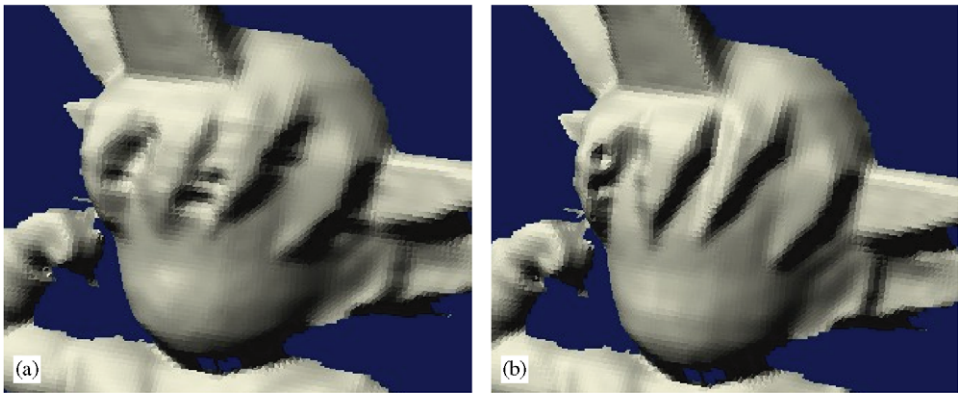


Fig. 20. Goku synthesized depth map: (a) obtained using the lower depth values; and (b) obtained using the depth values with lower distortion.

In Fig. 20(a), such distortion considerations are neglected, and the depth map $Z^{i\to *}$ with lowest value (i.e., closest to the viewer) is chosen for each sample. It can be seen that small details are missing around the "hair" of the cartoon figure; this is due to the influence of contributions which are subject to high distortion. This problem is solved by selecting the source $Z^i$ based on distortion information, as shown in Fig. 20(b). In this example, all source depth maps have the same compressed


Fig. 21. Rendering $V^*$ obtained with: (a) 0.4 bpp depth maps and 0.4 bpp images; (b) 0.025 bpp depth maps and 0.4 bpp images; (c) 0.4 bpp depth maps and 0.025 bpp images; and (d) 0.025 bpp depth maps and 0.025 bpp images.

bit-rate, so that the distortion-sensitive synthesis procedure is principally responding to local expansion in the mapping from different source depth maps to $Z^*$.

Finally, we want to show the result of mixing different quality images $V^i$ with different quality depth maps $Z^i$. This represents a preliminary attempt to evaluate what kind of information is more important for rendering, which is central to a good server distribution policy. As mentioned already, development of the server policy lies beyond the scope of this present paper; however, the results presented in Fig. 21 provide strong support for the importance of correctly distributing bandwidth between geometry and texture information. Fig. 21 shows the impact of selecting views $V^i$ and depth maps $Z^i$ with different compressed qualities. In each case, all eight $V^i$ are compressed with the same quality. Similarly, all eight $Z^i$ are compressed with the same quality as each other; synthesis is performed jointly over the full set of eight view/depth pairs. First of all, it appears that the best renderings are obtained with high-quality views, even with low-quality depth maps. With low-quality view images $V^i$, the quality of the depth maps does not significantly affect the result. On the other hand, when the view images have sufficiently high bit-rates, the quality of the depth maps does affect the quality of rendered details. For example, Goku's face is stretched and he smiles more than he should in Fig. 21(b). Assuming the availability of coarse depth and view information at the client, therefore, a server should initially devote most of its bandwidth to refining the available view images. Geometry refinements might be sent only near the end of the progressive transmission; indeed they might never be sent if the interactive client selects a different view. This conclusion agrees broadly with that reached in [4].

## 8. Discussion and conclusions

This paper represents a first step toward a novel approach to the interactive dissemination of compressed 3D scenes. The framework presented here also draws attention to a variety of important problems such as the optimal distribution of compressed bits between texture and geometry information, and non-linear approximation (not just sub-sampling) of the plenoptic function. As a convincing start in this direction, we have described mechanisms for estimating the distortion associated with rendering an intended view from a variety of compressed images with uncertain geometry and we have experimentally validated a client-side rendering algorithm which aims to minimize this distortion. Furthermore, we have shown that there is no need for the server to send an explicit model of the surface geometry. Instead, it is sufficient to work exclusively with a collection of views and depth maps taken from the different view points. To this end, we have proposed and experimentally demonstrated a novel distortion-based framework for synthesizing both geometry and texture information for the desired view, based on an arbitrary set of distorted source depth maps and views. This framework allows new view/depth information to be added to the server and/or the client at any time, for progressive and ongoing refinements to the 3D interactive browsing experience.

While the discussion of optimal service policies lies beyond the scope of the present paper, it is worth emphasizing the benefits of the proposed framework which serves to decouple the client's rendering policy from the server's distribution policy. The client's responsibility is to produce the

best possible rendering of any desired view, based upon the information currently available to it. This means that the client can "walk through" the scene even while disconnected from the server, using data acquired during a previous browsing session or perhaps forwarded by another client (e.g., by email attachment). When attached to the server, the client can hope to receive data which will enable it to progressively improve a rendered view of interest, but will also almost certainly improve the quality associated with other nearby views.

The server does not have any direct control over the way in which clients will use the incremental image enhancements it delivers, either for depth maps or for view images. However, by providing high-quality, relevant information, the server can expect that a good distortion-sensitive client rendering algorithm will be able to exploit it. If the client's view of interest coincides exactly with an original view available at the server, delivery of this view will eventually be the only way to continuously reduce the client's rendering distortion, without the impact of illuminant- or geometry-induced distortions, or interior holes. However, this is not generally the best way to start serving a recently changed view of interest; indeed, this *ideal* source view might never be transmitted if the user's interests change.

## References


[1] L. Ballan, N. Brusco, G. Cortelazzo, 3D passive shape recovery from texture and silhouette information, in: Second European Conference on Visual Media Production (CVMP05, London), November 2005.

[2] L. Balmelli, Rate-distortion optimal mesh simplification for communications, Ph.D. Dissertation, Ecole Polytechnique Federale de Lausanne, Switzerland, 2001.

[3] P. Burt, E. Adelson, A multiresolution spline with application to image mosaics, Trans. ACM Graphics 2 (4) (October 1983) 217–236.

[4] I. Cheng, A. Basu, Reliability and judging fatigue reduction in 3D perceptual quality, in: IEEE International Symposium on 3DPVT, September 2004.

[5] D. Cohen-Or, Model-based view-extrapolation for interactive VR web systems, in: Proceedings of the Computer Graphics International, June 1997, pp. 104–112.

[6] H. Hoppe, Progressive meshes, in: Proceedings of the Computer Graphics, Annual Conference Series, 1996, pp. 99–108.

[7] ISO/IEC 15444-1, Information technology—JPEG 2000 image coding system—Part 1: core coding system, 2000.

[8] ISO/IEC 15444-2, Information technology—JPEG 2000 image coding system—Part 2: extensions, 2002.

[9] ISO/IEC 15444-9, Information technology—JPEG 2000 image coding system—Part 9: interactivity tools, APIs and Protocols, 2004.

[10] A. Khodakovsky, P. Schroder, W. Sweldens, Progressive geometry compression, in: Proceedings of the SIGGRAPH, 2000, pp. 271–278.

[11] D. Koller, M. Turitzin, M. Levoy, M. Tarini, G. Croccia, P. Cignoni, R. Scopigno, Protected interactive 3D graphics via remote rendering, in: Proceedings of the SIGGRAPH, 2004.

[12] M. Levoy, Polygon-assisted JPEG and MPEG compression of synthetic images, in: Proceedings of the SIGGRAPH, vol. 3, August 1995, pp. 21–28.

[13] P. Ramanathan, B. Girod, Receiver-driven rate-distortion optimized streaming of light fields, in: Proceedings of the IEEE International Conference on Image Processing, vol. 3, September 2005, pp. 25–28.

[14] S. Rusinkiewicz, M. Levoy, QSplat: a multiresolution point rendering system for large meshes, in: Proceedings of the SIGGRAPH, 2000, pp. 343–352.

[15] A. Secker, D. Taubman, Highly scalable video compression with scalable motion coding, IEEE Trans. Image Process. 13 (8) (August 2004) 1029–1041.

[16] D. Taubman, Localized distortion estimation from already compressed JPEG2000 images, in: Proceedings of the IEEE International Conference on Image Processing, October 2006.

[17] D. Taubman, R. Prandolini, Architecture, philosophy and performance of jpip: internet protocol standard for JPEG 2000, in: International Symposium on Visual Communication and Image Processing, vol. 5150, July 2003, pp. 649–663.

[18] D. Tian, G. AlRegib, FQM: A fast quality measure for efficient transmission of textured 3d models, in: Proceedings of the ACM-Multimedia, October 2004, pp. 686–692.

[19] N. Brusco, D. Taubman, G. Cortelazzo, Greedy non-linear approximation of the plenoptic function for interactive transmission of 3d scenes, in: Proceedings of the IEEE International Conference on Image Processing, vol. 1, September 2005, pp. 629–632.