Adaptively synchronous scalable spread spectrum (A4S) data-hiding strategy for three-dimensional visualization

Khizar Hayat
COMSATS Institute of Information Technology
Department of Computer Science
University Road
Abbottabad, Pakistan

William Puech
University of Montpellier II
LIRMM Laboratory
UMR 5506 CNRS
161, Rue Ada
34392 Montpellier Cedex 05
France
william.puech@lirmm.fr

Gilles Gesquière
Aix-Marseille University
LSIS
UMR CNRS 6168
IUT, Rue R. Follereau
13200 Arles, France

Abstract. *We propose an adaptively synchronous scalable spread spectrum (A4S) data-hiding strategy to integrate disparate data, needed for a typical 3-D visualization, into a single JPEG2000 format file. JPEG2000 encoding provides a standard format on one hand and the needed multiresolution for scalability on the other. The method has the potential of being imperceptible and robust at the same time. While the spread spectrum (SS) methods are known for the high robustness they offer, our data-hiding strategy is removable at the same time, which ensures highest possible visualization quality. The SS embedding of the discrete wavelet transform (DWT)-domain depth map is carried out in transform domain YCrCb components from the JPEG2000 coding stream just after the DWT stage. To maintain synchronization, the embedding is carried out while taking into account the correspondence of subbands. Since security is not the immediate concern, we are at liberty with the strength of embedding. This permits us to increase the robustness and bring the reversibility of our method. To estimate the maximum tolerable error in the depth map according to a given viewpoint, a human visual system (HVS)-based psychovisual analysis is also presented.* © *2010 SPIE and IS&T.* [DOI: 10.1117/1.3427159]

1 Introduction

The first decade of the twenty-first century has witnessed a revolution in the form of memory and network speeds and computing efficiencies. Simultaneously, the platform and client diversity base have also been expanding, with powerful workstations on one extreme and handheld portable devices, like smart phones, on the other. Still versatile are the clients whose needs evolve with every passing moment. Hence, the technological revolution notwithstanding, dealing with the accompanying diversity is always a serious challenge. This challenge is more glaring in the case of applications where huge data are involved. One such application is the area of 3-D visualization where the problem is exacerbated by the involvement of more than one set of data.

A typical 3-D visualization is based on at least two sets of data: a 2-D intensity image, called texture, with a corresponding 3-D shape rendered in the form of a range image, a shaded 3-D model, and a mesh of points. A range image, also sometimes called a depth image, is an image in which the pixel value reflects the distance from the sensor to the imaged surface.[1] The underlying terminology may vary from field to field, e.g., in terrain visualization height/depth data are represented in the form of discrete altitudes which, upon triangulation, produce what is called a digital elevation model (DEM): the texture is a corresponding aerial photograph overlaid onto the DEM for visualization. Similarly, in 3-D facial visualization, the D color face image represents the texture but the corresponding depth map is usually in the form of what is called a 2.5-D image. The

Paper 09119RR received Jul. 10, 2009; revised manuscript received Mar. 22, 2010; accepted for publication Mar. 26, 2010; published online May 21, 2010.

1017-9909/2010/19(2)/023011/16/$25.00 © 2010 SPIE and IS&T.

latter is usually obtained[2] by the projection of the 3-D polygonal mesh model onto the image plane after its normalization.

To cater to client diversity and unify the disparate 3-D visualization data, we are proposing in this work an adaptively synchronous scalable spread spectrum (A4S) data-hiding strategy. We are relying on the multiresolution character of the DWT-based JPEG2000 format for scalability, whereas data hiding is being employed to unify the data into a single standard JPEG2000 format file. The proposed data-hiding method is blind, robust, and imperceptible. Robustness is offered by spread spectrum (SS) embedding, and imperceptibility is provided by the removable nature of the method.

One might wonder why we employ data hiding when a depth map can be easily inserted to the format header of JPEG2000 compressed texture. One can even think of considering the DEM data as the fourth component besides the YCrCb components of texture. Our counterargument is that although there exist such approaches, e.g., GeoJP2[3] and GMLJP2[4] for terrain visualization, the problem is the escalation in the size of the texture file. That is to say, the process is additive and the final coded texture is of the order $X+Y$, where X is the size of the original texture and Y is the size of its depth map. What we are after is that the coded texture should be on the order of X, i.e., $X+Y \approx X$. In addition, the data will be synchronized and each texture block/coefficient would roughly contain its own depth map coefficient, a feature lacking in the aforementioned approaches, and they may need explicit georeferencing to achieve it. Note that GeoJP2 is a GeoTIFF-inspired method for adding geospatial metadata to a JPEG2000 file. The GeoTIFF specification[5] defines a set of TIFF tags provided to describe all cartographic information associated with TIFF imagery that originates from satellite imaging systems, scanned aerial photography, scanned maps, and DEM. In itself, GeoTIFF does not intend to become a replacement for existing geographic data interchange standards. The mechanism is simple using the widely supported GeoTIFF implementations, but the introduction of new universally unique identification (UUID) boxes has the disadvantage that there is an increase in the original JPEG2000 file size. GMLJP2, on the other hand, envisages the use of the geography markup language (GML) within the XML boxes of the JPEG 2000 data format in the context of geographic imagery. A minimally required GML definition is specified for georeferencing images while also giving guidelines for encoding of metadata, features, annotations, styles, coordinate reference systems, and units of measure as well as packaging mechanisms for both single and multiple geographic images. DEMs are treated the same way as other image use cases, whereas coordinate reference system definitions are employed using a dictionary file. Thus DEM is either provided as a TIFF file and its name is inserted between proper GML tags, or its points are directly inserted into the GMLJP2 file. In the former case, there is no reduction in the number of files, whereas in the latter case the amount of data is increased.

In real-time face visualization context, two areas are of special interest in contemporary research, namely video conference and video surveillance. In earlier video conference applications, the aim was to change the viewpoint of the speaker. This allowed in particular the recreation of a simulation replica of a real meeting room by visualizing virtual heads around a table.[6] Despite the fact that many technological barriers have been eliminated—thanks to the availability of cheap cameras, powerful graphic cards, and high bitrate networks—there is still no efficient product that offers a true conferencing environment. Some companies, such as Tixeo in Montpellier, France[7] propose a 3-D environment where interlocutors can interact by moving an avatar or by presenting documents in a perspective manner. Nevertheless, the characters remain artificial and do not represent the interlocutors' real faces. In fact, it seems that changing the viewpoint of the interlocutor is considered more a gimmick than a useful functionality. This may be true of a video conference between two people, but in the case of a conference that would involve several interlocutors spread over several sites that have many documents, it becomes indispensable to replicate the conferencing environment. In the case of 3-D face data, wavelets have been extensively employed,[8] but rather than the visualization, the focus has traditionally been on feature extraction for face recognition.

The rest of the work is arranged as follows. Our method is explained with details in Sec. 2, while the results we obtained are elaborated on in Sec. 3. Section 4 concludes this work.

2 Proposed Adaptively Synchronous Scalable Spread Spectrum Data-Hiding Method

In this section, we present our method for adaptive scalable transfer and online visualization of textured 3-D data. In our previous efforts, a LSB-based data-hiding strategy was employed to synchronously unify lossless wavelet-transformed DEM in the Y/Cr/Cb plane of the corresponding texture in the DWT domain from the JPEG2000 pipeline.[9] In the said work, the focus was perceptual transparency and little attention was paid to robustness. That is why LSB-based embedding was employed. This present effort is different from the previous one in various aspects. First, we are opting for a spread spectrum (SS) strategy for improving robustness, especially with communication across some noisy channel. We know that LSB-based embedding is the most vulnerable to noise and there is every chance to lose vital depth map information. Besides, by opting for SS embedding we have not lost sight of perceptual transparency, as the reversibility of our strategy enables us to get the maximum possible texture quality. The second improvement, from the past, is the adjustment in the synchronization in embedding. Rather than synchronizing with a whole component, we are adapting it to a subset of subbands of the component. This would enable us to have far better quality of the depth map than the past. Closely related is the systematic calculation in the form of the level of detail (LOD) viewpoint analysis, carried out in the following section to determine allowable error in the depth map. The employment of wavelet-based strategies can be found in the literature, some of which may be of interest to the reader. Royan *et al.*[10] have utilized the wavelet-based hierarchical LOD models to implement view-dependent progressive streaming for both client-server and P2P networking scenarios. In an earlier work, Gioia, Aubaut, and Bouville proposed to reconstruct wavelet-encoded large

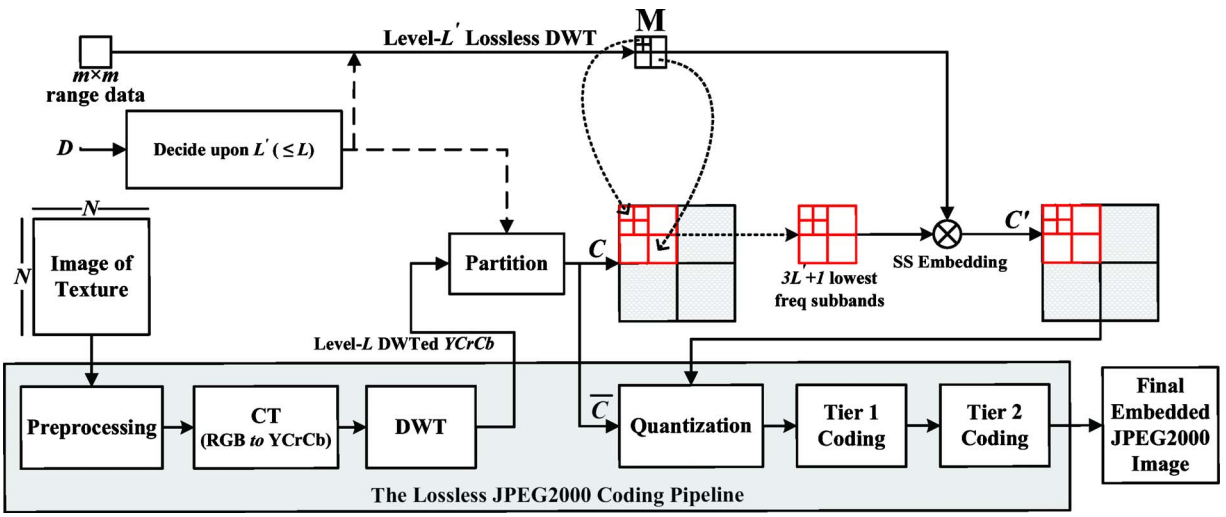

Fig. 1 Global overview of the encoding process.

meshes in real time in a view-dependent manner for visualization by combining local update algorithms, cache management, and client/server dialog.[11] A good and advanced application of wavelets to real-time data for terrain visualization can be found in the work of Kim and Ra.[12] The authors rely on restricted quad-tree triangulation for surface approximation. The focus of their work is, however, the third dimension, i.e., DEM, and suffers from a lack of integration with the texture.

For the embedding method, we are opting for data hiding to unify disparate visualization data, but it must be noted that we are not employing data hiding for copyright application nor concealing the existence of a message (steganography). Of the data-hiding techniques, most likely the LSB-based methods offer higher perceptual transparency as far as blind nonremovable embedding is concerned. This is, however, accompanied by a marginal loss of data. Therefore, to get the maximum in terms of perceptual transparency and robustness, we are opting for a different blind embedding strategy that may not necessarily be nonremovable. To this end, it is proposed here to employ a SS data-hiding strategy pioneered by Cox, Miller, and Bloom.[13] The SS methods offer high robustness at the expense of cover quality, but this quality loss is reversible, since the embedded data can be removed after recovery. The proposed adaptive synchronization is helpful in improving the quality of the range data approximation for a given texture approximation. The range data error can thus be reduced at the expense of texture quality, but since the data-hiding step is reversible, the highest possible texture quality is still realizable.

An overview of the method is described in Sec. 2.1. Section 2.1 depicts the viewpoint analysis for the adaptation of synchronization, while the embedding and the decoding processes are explained in Secs. 2.2 and 2.3, respectively. The reconstruction procedure is outlined in Sec. 2.4.

2.1 *Overview of the Method*

Suppose an $N\times N$ texture image has a depth map of $m\times m$ coefficients. In the spatial domain, let each of the coefficients correspond to a $t\times t$ pixel block of the related texture, where $t=N/m$. Suppose the texture is to be JPEG2000 coded at DWT decomposition level L, implying $R=L+1$ resolutions. Let us apply lossless DWT to the range coefficients at level L', where $L'\leqslant L$.

For embedding, we interrupt the JPEG2000 coding of the texture after the DWT step, as illustrated in Fig. 1, and use one of the transformed YCrCb components as a carrier for embedding. The carrier (C) is partitioned into $m\times m$ equal-sized block, $B_{i,j}$ with size dependent on the value of L'. If $L'=L$, then C consists of the whole of the selected component(s) and embedding block size remains $t\times t$, since no subband is excluded from the possible data insertion. Otherwise, for $L'<L$, only a subset of subbands—the lowest $3L'+1$ of the original $3L+1$ after excluding the remaining $3(L-L')$ higher frequency subbands—constitute C, and $B_{i,j}$ has a reduced size of $t/2^{(L-L')}\times t/2^{(L-L')}$. Care must be taken of the fact that block size must be large enough to reliably recover the embedded data after correlation. This decision about the choice of L' is especially important in the case of terrain visualization, where a viewpoint analysis may be quite useful to ascertain the maximum tolerable error in the DEM that would in turn determine the extent to which the synchronization must be adapted. Hence the important factors in reaching a decision are based on the value of L', the block size, and the involvement or otherwise more than one YCrCb component in embedding. Our strategy is aimed at the best 3-D visualization as a function of the network connection and the computing resources of the client. The proposed adaptive approach to embed the range data in the DWT texture is a function of the distance D between the viewpoint and the depth map. The latter's quality is evaluated with the root mean square error (RMSE) in length units. As pointed out in some recently published works,[14] even today the acquisition of the DEM for 3-D terrain visualization is error prone and it is difficult to get a RMSE less than 1 m. To calculate the maximum acceptable RMSE for an optimal 3-D visualization, we rely on the distance D between the viewpoint and the depth map, as illustrated in Fig. 2, and the visual acuity (VA) of the human visual system (HVS).

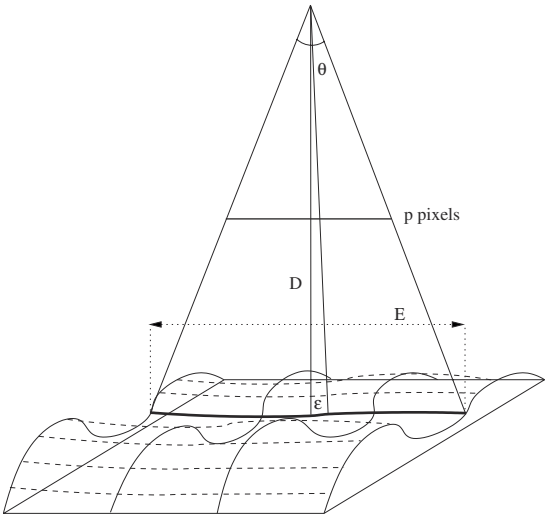

Fig. 2 Visualization of a depth map from a viewpoint.

Visual acuity is the spatial resolving capacity of the HVS. It may be thought of as the ability of the eye to see fine details. There are various ways to measure and specify visual acuity, depending on the type of acuity task used. VA is limited by diffraction, aberrations, and photoreceptor density in the eye.[15] In this work, for the HVS we assume that the VA corresponds to an arc θ_{VA} of 1 min ($1'=1/60$ deg). Then for a distance D, the level of detail (LOD) is:

$$\mathrm{LOD}=2\times D\times\tan(\theta_{\mathrm{VA}}). \tag{1}$$

For example, in the case of terrain visualization, if $D=3$ m then $\mathrm{LOD}=87\times10^{-4}$ m, and if $D=4$ km then $\mathrm{LOD}=1.164$ m. For our application, illustrated in Fig. 2, if we want to see all the depth map, we have a relation between D and the size of the range data (width$=E$ m):

$$E=2\times D\times\tan(\theta/2). \tag{2}$$

Usually the field of view of the HVS is $\theta=60$ deg. For example, if $E=3200$ m, then $D=2771.28$ m.

For 3-D visualization, we should take into account the resolution of the screen in pixels. As illustrated in Fig. 2, if we have an image or a screen with p pixels for each row, then the LOD ϵ in p is:

$$\epsilon=\frac{E/2}{\tan(\theta/2)}\times\tan(\theta/p). \tag{3}$$

With $\theta=60$ deg, $E=3200$ m, and a resolution of 320 pixels (for a PDA, for example), we have a LOD $\epsilon=9.07$ m. Then, in this context, for our application we can assume that a RMSE near 9 m is acceptable for the DEM in the case of terrain visualization. Notice that we generate a conservative bound by placing an error of the maximum size as close to the viewer as possible with an orthogonal projection of the viewpoint on the depth map. We also assume that the range data model is globally flat, and that the accuracy between the center and the border of the depth map is the same. In reality, the analysis should be different, and part of the border should be cut as explained by Ref. 16 in a particular case of a cylinder. Anyhow, the value of D would help us to reach a decision about the value of L'.

2.2 *Embedding Step*

The process of embedding is made in a given DWT-domain range data coefficient of the carrier C, which is one or more of the YCrCb components of the transformed texture. The criteria for ascertaining the carrier depends on L', t, and ω, where ω is the number of bits assigned to represent a single DWT-domain range data coefficient. If T is the threshold on the number of carrier coefficients needed to embed a single bit, then algorithm 1 proposes a solution to choose only one or three components.

Algorithm 1. A subroutine to choose component (s) for embedding.

1. **If** $(t^2/2^{2(L-L')}\geqslant T\omega)$ **then**
2. **select** for C one of the three YCrCb components
3. **else**
4. **if** $(T\omega>t^2/2^{2(L-L')}\geqslant T\omega/3)$ **then**
5. C is constituted by the three YCrCb components
6. **else**
7. A4S embedding may not be convenient
8. **end if**
9. **end if**.

The choice whether to embed in the Y or Cr/Cb plane depends on the fact that Y plane embedding would distort the encoded image, while chrominance plane embedding would escalate the final file size. Neither is the former a serious issue, as embedding is removable, nor is the latter, since it may matter only when $L'=L$.

The embedding process of a DEM coefficient in a given block $(\mathrm{size}=t^2/2^{2(L-L')})$ is elaborated by the flowchart given in Fig. 3. Since one coefficient is embedded per $t/2^{(L-L')}\times t/2^{(L-L')}$ block, each $B_{i,j}$ is repartitioned into as many subblocks (b_k) as there are number of bits used to represent a single transformed coefficient. Each of b_k from $B_{i,j}$ would carry the k'th bit β_k of the transformed range altitude $d_{i,j}$. Embedding depends on a key generating a pseudorandom matrix $\mathbf{W}$ with entries from the set $\{\alpha,-\alpha\}$. The matrix $\mathbf{W}$ has the same size as b_k, i.e., $\mathbf{W}=[y_{uv}]_{n\times n}$, where $y_{uv}\in\{\alpha,-\alpha\}$. The scalar α is referred to as the strength of embedding. If the bit to embed β_k is 1, then matrix $\mathbf{W}$ is added to the matrix b_k, otherwise $\mathbf{W}$ is subtracted from b_k. The result is a new matrix b'_k that replaces b_k as a subblock in the embedded block $B'_{i,j}$ of the marked carrier C'.

Two factors are important here. First is the DWT level (L') of transformed range data before embedding, which is a tradeoff between the final texture quality and its range data quality. At the decoding end, the quality of the range data would depend on the difference between L and L'. The larger the difference $(L-L')$, the higher the quality will be

and vice versa. Second is the value of α, since a larger α means high degradation of the watermarked texture. This second factor is, however, of secondary importance since the embedded message (M) after recovery will be used to correct any loss in texture quality. So, no matter how much degradation is there, the reconstructed texture should be of the highest possible quality.

2.3 *Decoding Step*

The prior coded image can be utilized like any other JPEG2000 image and sent across any communication channel. The decoding is more or less converse to the previous process. Just before the inverse DWT stage of the JPEG2000 decoder, the range data can be extracted from C' using the previously mentioned partitioning scheme, i.e., $B'_{i,j}$ blocks and their b'_k subblocks. Figure 4 shows the flowchart for the recovery of a range altitude $d_{i,j}$ from subblocks b'_k of $B'_{i,j}$ and eventual reconstruction of $b_{i,j}$, and ultimately $B_{i,j}$. A given k'th subblock b'_k and also the matrix $\mathbf{W}$ can be treated as a column/row vector, and the Pearson's correlation coefficient[17] ρ can be computed. If ρ is closer to -1 than 1, then the embedded bit β was a 0, otherwise it was a 1. Once this is determined, then it is obvious that what were the entries of b_k, i.e., if β is 0, then add $\mathbf{W}$ to b'_k, otherwise subtract $\mathbf{W}$ from b'_k.

Algorithm 2. Embedding of the range data coefficient $d_{i,j}$ in the corresponding block $B_{i,j}$.

1. **Begin**
2. **get** the (i,j)'th partition $B_{i,j}$ of the cover and the corresponding 16 bit coefficient $d_{i,j}$
3. **partition** $B_{i,j}$ to 16 subblock, $b_0, b_1, \ldots b_{15}$
4. **for** $k \leftarrow 0$ to 15 **do**
5. **read** the k'th bit β_k of the DEM coefficient $d_{i,j}$
6. **if** $\beta_k = 0$ **then**
7. **set** $b'_k \leftarrow b_k - \mathbf{W}$/*$\mathbf{W}$ is a one-time pseudorandomly constructed matrix*/
8. **else**
9. **set** $b'_k \leftarrow b_k + \mathbf{W}$
10. **end if**
11. **replace** b_k by b'_k in the block $B_{i,j}$, which will ultimately change to $B'_{i,j}$
12. **end for**
13. **replace** $B_{i,j}$ by $B'_{i,j}$ in the cover
14. **end**.

2.4 *Reconstruction Step*

Now comes the reconstruction phase, where by the application of 0-padding, one can have $L+1$ and $L'+1$ different approximation images of texture and their range data, respectively. This is where one achieves the scalability goal. Our method is based on the assumption that it is not necessary that all the subbands are available for reconstruction, i.e., only a part of C' is on hand. This is one of the main advantages of the method, since the range data and the texture can be reconstructed with even a small subset of the coefficients of the carrier. The resolution scalability of wavelets and the synchronized character of our method enable a 3-D visualization, even with fewer than original resolution layers as a result of partial or delayed data transfer. The method thus enables us to effect visualization from a fraction of data in the form of the lowest subband of a particular resolution level, since it is always possible to stuff 0s for the higher bands. The idea is to have a 3-D visualization utilizing lower frequency subbands at level L'', say, where $L'' \leqslant L$. For the rest of $L-L''$, part one can always pad a 0 for each of their coefficients. The inverse DWT of the 0-stuffed transform components will yield what is known as an image of approximation of level L''. Before this, as depicted by algorithm 3, data related to the third dimension, i.e., range data, must be extracted whose size depends on both L'' and L'. Thus if $L' \leqslant L''$, one will always have the entire set of the embedded altitude, since all of them will be extractable. We would have a level 0 approximate final range map after inverse DWT of the highest possible quality. On the other hand, if $L' > L''$, one would have to pad 0s for all coefficients of higher $L'-L''$ subbands of a transformed range map before inverse DWT, which would result in a level L''-approximate range data of an inferior quality.

Algorithm 3. The reconstruction process via 0-padding.

1. **Begin**
2. **read** the coded texture data corresponding to level-L'' subbands
3. **decode** to extract the range data that may correspond to L'' or a larger level $L''+\Delta$, where $\Delta \leqslant 0$ depends on the extent of adaptation
4. **for** both the texture and extracted range data **do**
5. **initialize** c_l to L'' if dealing with texture or to $L''+\Delta$, otherwise
6. **stuff** a 0 for every coefficient of the subbands with level $\neq c_l$
7. **apply** inverse DWT to the result
8. **add** the result with the (c_l-1) approximation to get the (c_l) approximation
9. **end for**
10. **end**.

3 Experimental Results and Analyses

We have applied our method to examples from two areas, namely terrain and facial visualizations, with the results presented and analyzed in Secs. 3.1 and 3.2, respectively. In Sec. 3.3, we take the allowable limit of adaptation and apply the method to a set of 3-D texture/range data pairs, and analyze the robustness offered by our method.

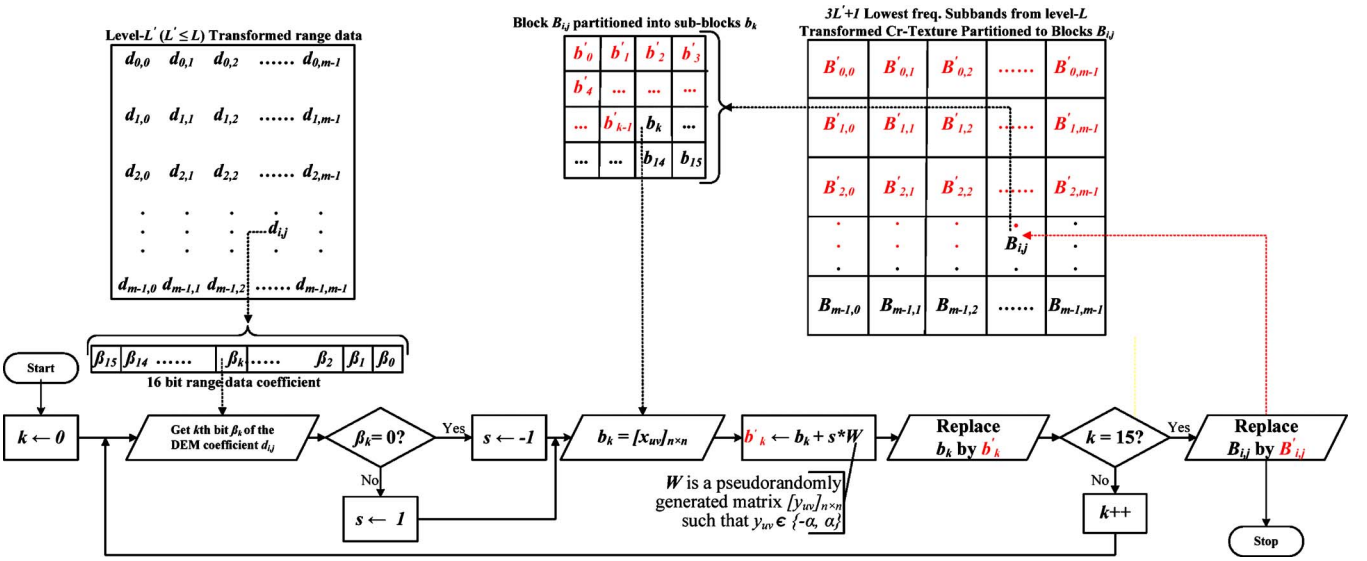

Fig. 3 Flowchart of the DWT domain SS embedding of the depth coefficient $d_{i,j}$ in the block $B_{i,j}$ of the lowest $3L'+1$ texture subbands.

3.1 *Terrain Visualization*

We have chosen the example texture/DEM pair given in Fig. 5 to try various possible adaptations in embedding. We are using a tiling approach: each DEM and texture are decomposed in small parts to facilitate the transfer between the server and clients. The DEM [Fig. 5(a)] is shown in the form of a 32×32 grayscale image, where the whiteness determines the height of the altitude. The corresponding texture [Fig. 5(b)] has a size of 3072×3072 pixels, implying one DEM coefficient per 96×96 texture block, i.e., $t=96$. For the purpose of comparison, a 256×256 pixel portion, at (1000, 1500) coordinates, is magnified as shown in Fig. 5(c). Figure 5(d) illustrates a 3-D view obtained with the help of the texture/DEM pair. We chose to subject the texture to reversible JPEG2000 encoding at $L=4$ that would give us five possible resolutions (13 subbands) based on 1, 4, 7, and 10 lowest frequency or all of the 13 subbands.

For fully synchronous embedding, all 13 subbands of the selected component were utilized in embedding, and thus the DEM was subjected to lossless DWT at level $L'=L=4$ to give 13 subbands. The embedding block $B_{i,j}$ then had a size of 96×96 for the embedding of one 16-bit transformed DEM coefficient, which means a 24×24 subblock (b_k) per DEM bit. Since b_k is large enough, the needed strength/amplitude (α) of embedding is smaller and the quality of the luminance/chrominance component will not deteriorate much. For our example, it was observed that 100% successful recovery of the embedded message is realized when $\alpha=2$ for embedding chrominance component (Cr/Cb), as against $\alpha=8$ for embedding luminance component (Y). The overall quality of the watermarked texture was observed to be better (44.39 dB) for Cr/Cb than for Y (26.53 dB). This quality difference is not that important, however, since the original texture is almost fully recoverable from the watermarked texture. On the other hand, there is a risk that embedding in Cr/Cb may eventually inflate the size of the coded image. This risk was, however, not that important for this example, since the observed bitrate was found to be only marginally escalated in the fully synchronized case—14.81 bpp compared to 14.55 bpp for any adaptation. The texture bitrate is thus independent of the extent of adaptation in partially synchronized cases.

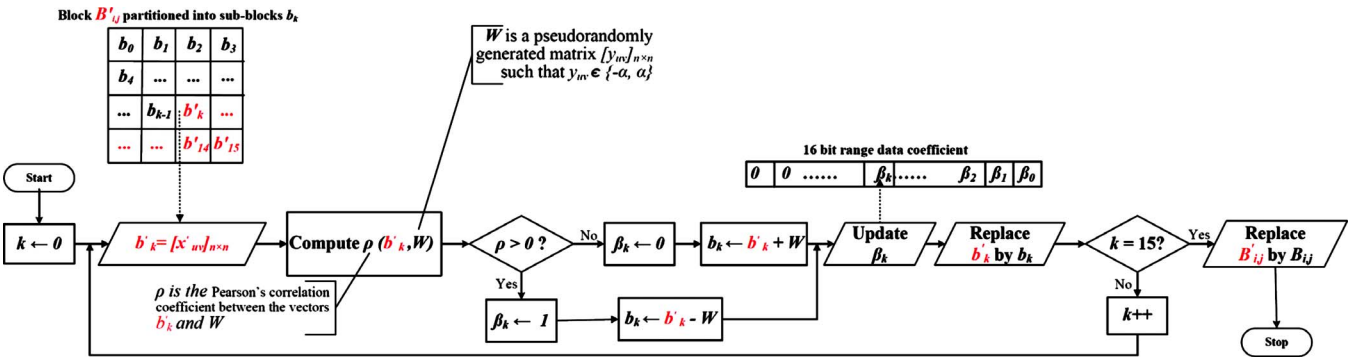

Fig. 4 Flowchart of the recovery of the depth coefficient $d_{i,j}$ and the block $B_{i,j}$ from the SS embedded block $B'_{i,j}$ of the lowest $3L'+1$ texture subbands.

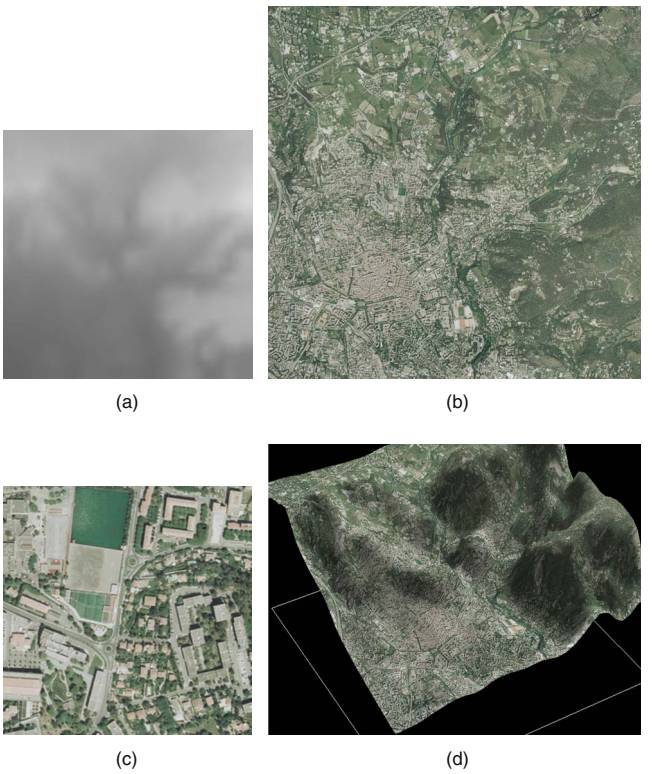

Fig. 5 Example images: (a) 32×32 DEM, (b) 3072×3072 pixel texture, (c) a 256×256 pixel detail of (b), (d) a corresponding 3-D view.

This is probably due to the exclusion of the three largest subbands from embedding; these bands, incidentally, have the highest probability of zero coefficients.

On the reintroduction of marked Y or Cr/Cb to the JPEG2000 pipeline, we get our watermarked texture image in JPEG2000 format. From this image one can have five different approximation images for both the DEM and texture. The level-l ($\leq L$) approximate image is the one that is constructed with $(1/4^l)\times 100\%$ of the total coefficients that correspond to the available lower $3(L-l)+1$ subbands. For example, a level-0 approximate image is constructed from all the coefficients, and a level-2 approximate image is constructed from 6.12% of the count of the initial coefficients. Since the embedded data are removable, one gets the highest possible qualities for all the texture approximations but not for the DEM, as its quality depends on our embedding strategy. Figure 6(a) shows the variations in texture quality as a function of the level of approximation. Since we have been able to extract the texture coefficients with 100% accuracy, any quality loss observed after that is not due to the proposed method, but external factors like the nature of texture, or types of wavelet employed by the JPEG2000 codec.

The sensitive nature of DEM compels us to avoid too much loss in its quality. For improved DEM quality, we had to adjust the synchronization, and rather than persisting with $L'=4$, we went for $L'<4$, which meant exclusion of $3(4-L')$ highest frequency subbands of the carrier from embedding; the synchronization was now maintainable between all the $3L'+1$ subbands of DEM and the subset $3L'+1$ lowest subbands of the carrier. Obviously, dimensions of $B_{i,j}$ and b_k were dyadically reduced by a factor of $2^{4-L'}$, which led to an increase in the value of α and an eventual degradation in the quality of the coded texture. As described in Sec. 2.3, the most important step of our approach is to adapt the synchronization. In other words, the objective of the step is to find the lower bound for L'. This bound depends on the texture-to-DEM size ratio and also on how much distortion in the carrier is reversible, i.e., bounds of α. For our example, the lowest quality for the coded texture—13.40 dB with $\alpha=64$ for the Y carrier [Figs. 7(a) and 7(b)] and 31.47 dB with $\alpha=16$ for the Cr carrier [Figs. 7(c) and 7(d)]—was observed at $L'=1$. The reason is that all the information had to be embedded in just

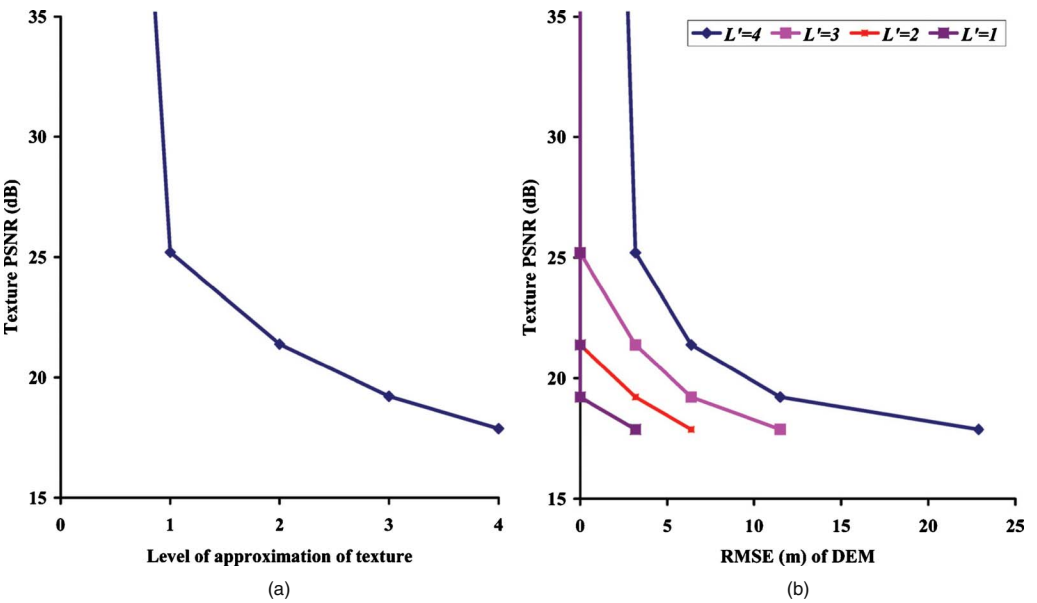

Fig. 6 Texture quality as a function of: (a) level of approximation of texture and (b) RMSE of DEM in meters.

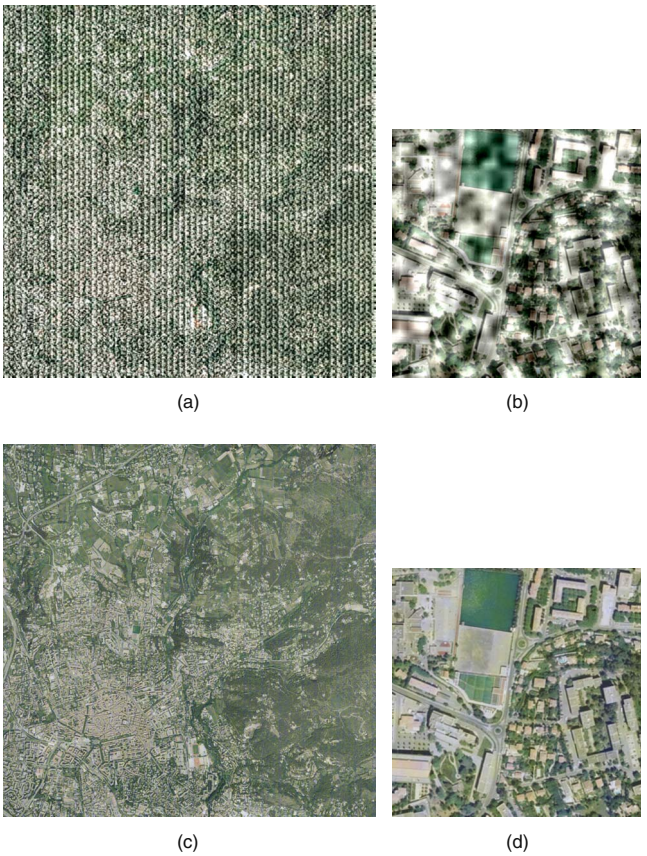

Fig. 7 Texture after embedding in: (a) Y plane ($\alpha=64$) with PSNR =13.40 dB, (b) its 256×256 detail, (c) Cr plane ($\alpha=16$) with PSNR=31.47 dB, and (d) its 256×256 detail.

four lowest energy and smallest subbands, implying a size of 3×3 for b_k with $\alpha=64$ for the Y carrier and $\alpha=16$ for a Cr/Cb carrier. Beyond this bound ($L'=0$, i.e., spatial domain), the block size did not allow for the recovery of the embedded message. Figure 6(b) illustrates the trend in texture quality as a function of the DEM error for various possible values of L'. To judge the DEM quality, root mean square error (RMSE) in meters (m), as explained in Sec. 2.1, was adopted as a measure. It can be observed that for the same texture quality, one can have various DEM qualities depending on L', the level of DEM decomposition before embedding. The smaller the L', the higher the degree of adaptation in synchronization and hence higher is the resultant DEM quality. With our approach, by using the example given in Fig. 5, the RMSE of the five possible DEM approximations, i.e., levels 0, 1, 2, 3, and 4, were found to be 0, 3.18, 6.39, 11.5, and 22.88 m, respectively. For full synchronization, as presented in Ref. 9, the worst DEM quality, 22.88 m, results when one goes for a 3-D visualization from level-four approximate texture, as the corresponding DEM will also be four-approximate. But even with one step adjustment, this quality is twice improved, and for $L'=3$, both the three and four approximate texture images have three approximate DEM with RMSE =11.5 m. Go a step further, and the maximum DEM error will be reduced to 6.39 m. Hence with adaptive synchronization, one can have a high quality DEM even for very low quality texture or, more precisely, the same quality DEM

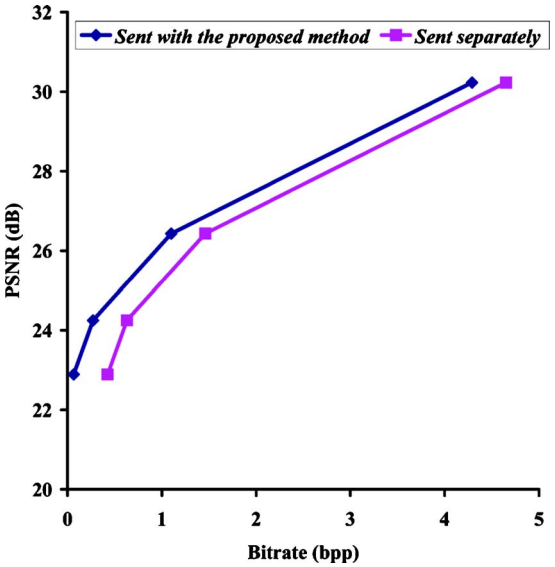

Fig. 8 Comparison between the proposed method and separate coding for the variation in texture quality as a function of its bitrate for a DEM error of 3.18 m.

for all the approximation images at levels $\geqslant L'$. Figure 8 shows the trend in texture quality for a given DEM error (3.18 m) as a function of bitrate. For the sake of comparison, we also show the graph when the texture and its DEM are coded separately. A level-one approximate separately encoded DEM corresponds to 0.36 bpp, and this is exactly the escalation for each point on the graph. In other words, for the example in hand, our method can give an advantage of 0.36 bpp in bitrate for a given PSNR. Note that we can have a texture/DEM pair with even a bitrate as low as 0.06 bpp and even 0.27 bpp with our method. Images corresponding to a DEM error of 3.18 m are shown in Fig. 9 and the resultant 3-D visualizations are illustrated in Fig. 10. It can be seen that with even a tiny fraction of the total coefficients [as low as 0.40%, i.e., Figs. 9(h) and 9(i), and Fig. 10(e)], a fairly commendable visualization can be realized.

The maximum error in RMSE of the DEM tolerable by an observer is a function of the distance of the observer, i.e., the viewpoint D. This threshold is mainly dependent on the human visual system (HVS) and for this reason, the analysis given in Sec. 2.1 is extremely useful to adapt the synchronization.

3.2 *Application to Three-Dimensional Face Visualization*

For comparable dimensions of the texture and depth map, we opt for a face visualization example. In this context, we applied our method to a set of examples of more 100 models from the FRAV3D[18] database, each consisting of a 512×512 face texture and a corresponding 64×64 depth map of 8-bit coefficients. Two examples are given in Fig. 11. Thus one depth map coefficient corresponds to an 8×8 texture block, which means that the margin for adaptation is narrow. Hence, even if we involve all the three planes (i.e., YCrCb) in embedding, we can adapt, at most, by one level with a SS embedding. The 8-bit depth map

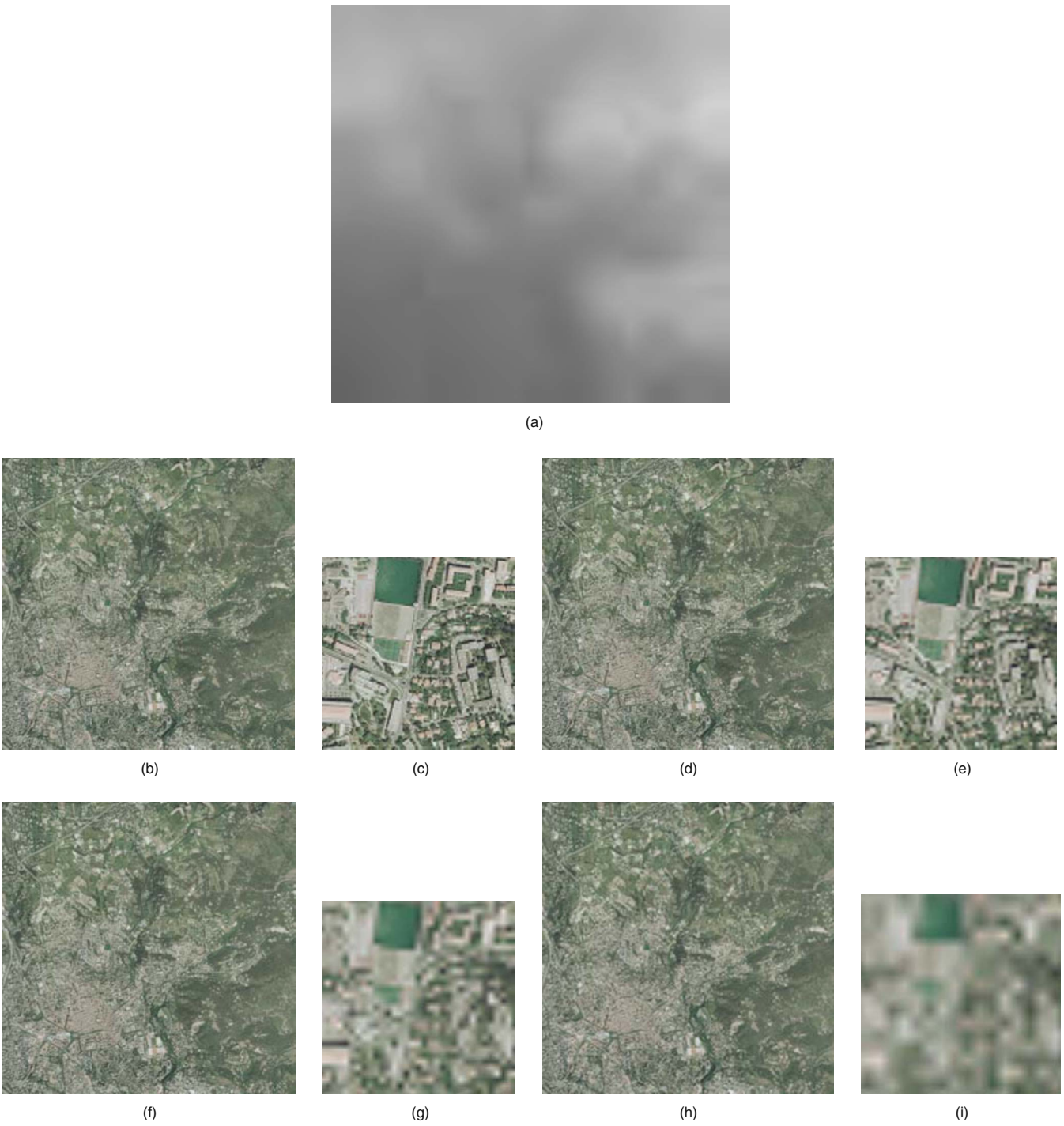

Fig. 9 Approximations corresponding to the 3.18-m error DEM ($L-1$ approximate): level-L' DEM is embedded in the lowest $3L'+1$ DWT domain subbands of level-four Cr-texture. (a) $L-1$ DEM, (b, c) $L-1$ approximate texture (4.29 bpp) with $L'=4$, (d, e) $L-2$ approximate texture (1.1 bpp) with $L'=3$, (f, g) $L-3$ approximate texture (0.27 bpp) with $L'=2$, (h, i) $L-4$ approximate texture (0.066 bpp) with $L'=1$.

was subjected to level-L' lossless DWT before embedding in the three DWT-domain components from the JPEG2000 coder. To ensure accuracy, we expand the word size for each of the transformed depth map coefficients by one additional bit, and represent it in 9 bits that are then equally distributed among the three planes for embedding. Taking $L=4$ would imply $L'=4$ for synchronous embedding, and $L'=3$ for a one-level adaptation in synchronous embedding. To embed one bit, the former would manipulate about 21 while the latter about five coefficients of Y, Cr, or Cb plane in the DWT domain.

For the set of face examples, the maximum of the lowest value of α needed for the 100% recovery of the embedded data was found to be 26 for the synchronous case and 45

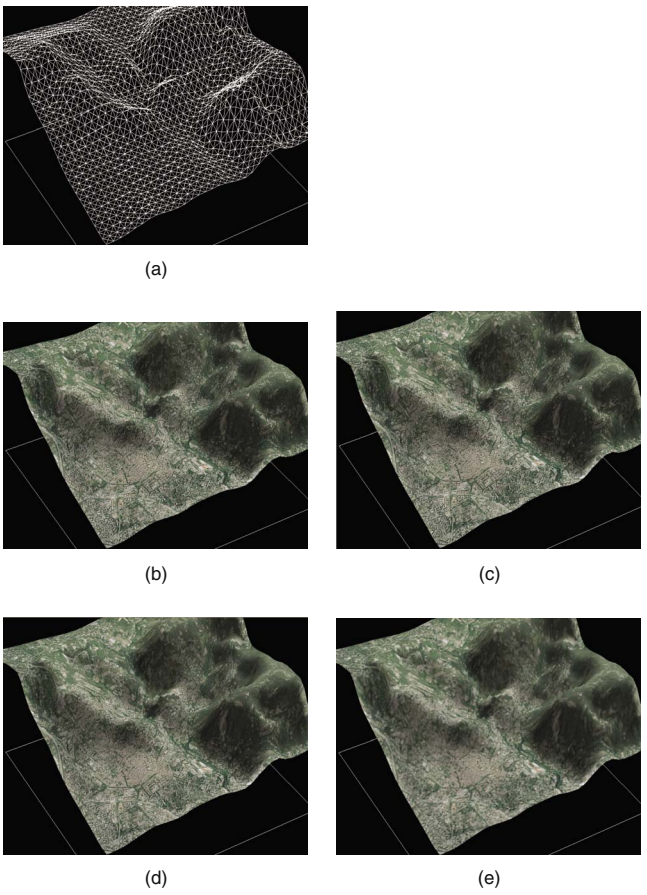

Fig. 10 3-D visualization corresponding to Fig. 9. (a) $L-1$ approximate DEM-3.18-m error, (b) $L-1$ approximate texture ($L'=4$)—4.29 bpp, (c) $L-2$ approximate texture ($L'=3$)—1.1 bpp, (d) $L-3$ approximate texture ($L'=2$)—0.27 bpp, and (e) $L-4$ approximate texture ($L'=1$)—0.066 bpp.

for the adaptive synchronous case. The resultant watermarked JPEG2000 format image was degraded to a mean PSNR of 17.15 dB for the former and a mean PSNR of 15.16 dB for the latter. Even with that much distortion, the reversible nature of the method enabled us to fully recover the embedded data on one hand and achieve the maximum possible quality in case of the texture, i.e., with a PSNR of infinity, on the other. Figure 12(a) plots the PSNR against the level of approximation for the reconstruction of texture. It can be seen that even a texture reconstructed with as low as 0.39% of the transmitted coefficients has a PSNR of around 30 dB. The advantage of adaptation is evident from the fact that, as shown in Fig. 12(b) for the same texture quality, even one-level adaptation offers far better depth map quality compared to full synchronization.

The evident edge of the adaptive strategy in depth map quality is conspicuous by the comparison of 3-D visualizations illustrated in Figs. 13 and 14. The latter is always better by one level (four times, roughly) as far as depth map quality is concerned. Hence with the same number of transmitted coefficients, one gets the same level-l approximate texture for the two cases, but the one-level adaptive case would have a level-$l-1$ approximate depth map compared to the relatively inferior level-l approximate depth map for the fully synchronized case.

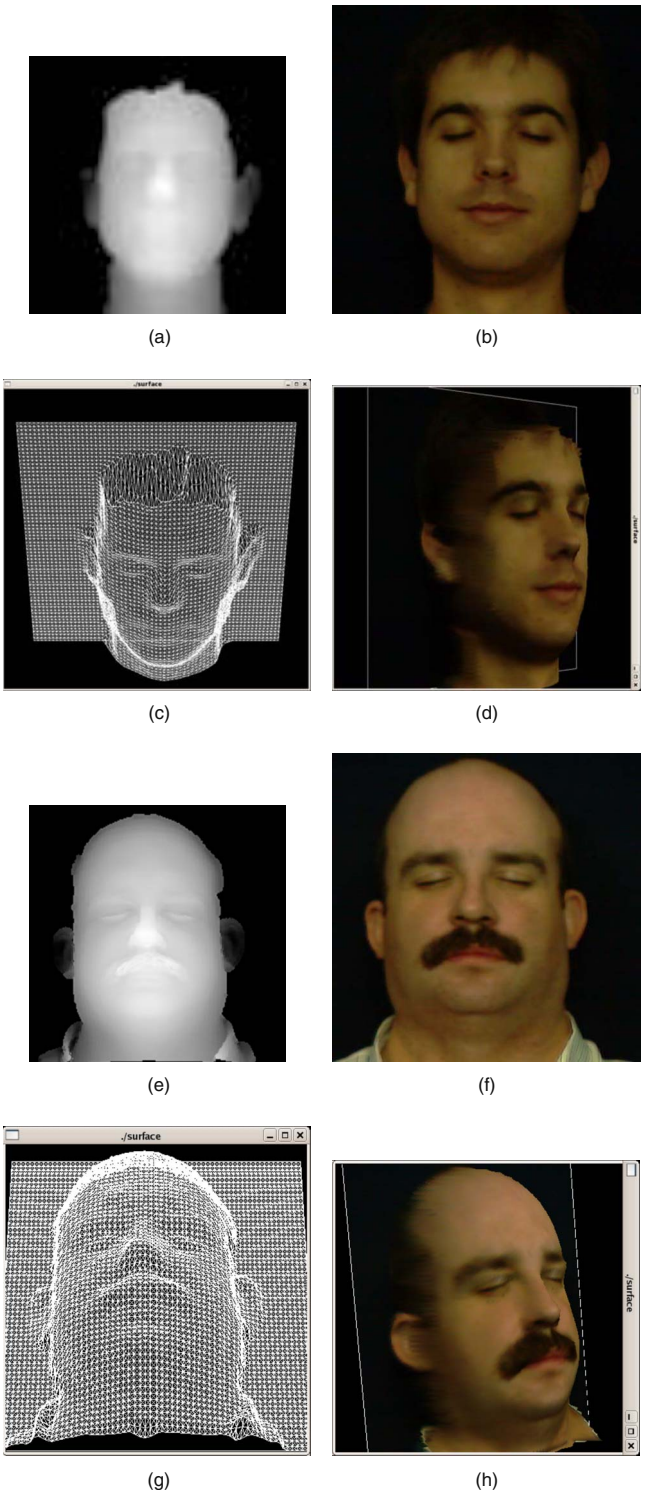

Fig. 11 Two examples of face visualization: (a) and (e) 64×64 depth maps, (b) and (f) 512×512 pixel face textures, (c) and (g) the corresponding 3-D meshes, and (d) and (h) 3-D views.

3.3 *Robustness Analysis*

The main purpose of employing an SS strategy was to improve the robustness of the embedded texture against noisy transmission and image manipulation. In the latter case, attacks are typically aimed at the change of image format, e.g., to JPEG, and cropping. We believe that our method

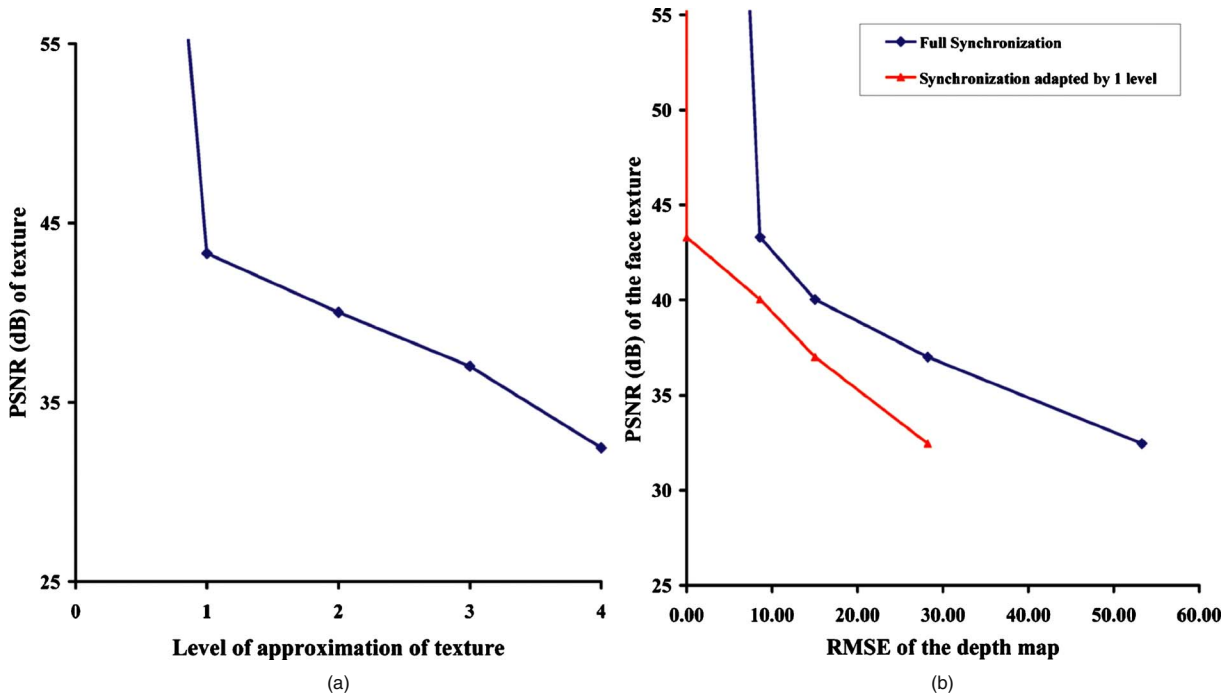

Fig. 12 Variation in the texture quality as a function of: (a) its approximation level and (b) RMSE of the extracted depth map.

gives us the sought-after robustness. In this section we are subjecting the marked texture to various manipulations to ascertain the claimed robustness. For the sake of brevity, we use the nomenclature described in Table 1 to represent various manners of the DWT domain embedding of the range data in the texture, e.g., desynY implies DWT domain embedding of range data coefficients in the Y plane of texture, excluding the three highest frequency subband coefficients (75% of the total).

Our method was applied to a set of 300 tiles corresponding to the region of *Bouches du Rhône* in France provided by IGN,[19] Paris, France. All the tiles were composed of a 1024×1024 pixel texture and a corresponding DEM of 32×32 16-bit altitudes. All the textures were subjected to level-four ($L=4$) decomposition in the JPEG2000 codec. The DEM were DWTed at level $L'=4$ for synY and synCr, and $L'=3$ for desynY and desynCr. For each pair we determined the minimum of α needed for 100% recovery of the embedded coefficients. For all such α's, we determined the minimum ($\alpha_{\min}$) and maximum ($\alpha_{\max}$) in the set images for each case, as given in Table 2. It can be observed from the PSNR values given in the table that the Y-plane embedding and larger α distort the texture more. Although the embedded message is fully removable, we still need to maintain the quality of the encoded texture, and that is why we settled for different $\alpha_{\max}$ for each case. Otherwise, the global maximum $\alpha_{\max}=49$ would have been a better choice. Hence, for the sake of comparison, we then carried out embedding in the example set of textures at $\alpha_{\max}$ conforming to Table 2 for each of the four embedding cases. The watermarked JPEG2000 coded textures were then subjected to robustness attacks.

3.3.1 *Resistance to JPEG compression*

To show the intensity of the JPEG compression, we have plotted the average PSNR of the distorted texture against the quality factor of compression, as shown in Fig. 15(a). It can be observed that for all four cases, the trend is identical, although the desynY case performs comparatively better at higher qualities. But since the embedding is removable, it is the integrity of the embedded data, i.e., DEM, which is important and that is why we have plotted the average bit error rate (BER) of the recovered DEM as a function of the JPEG quality factor in Fig. 15(b). The reason is that the abscissa going beyond 100% is for the purpose of representing the no-attack case, which must have a BER of zero. It can be observed that cases most sensitive to JPEG compression are the cases where embedding was carried out in the chrominance plane (desynCr, synCr), and even 100% quality factor of compression can result in significant errors. In contrast, Y-plane embedding offers high robustness, with desynY being the most robust.

3.3.2 *Robustness against Gaussian noise addition*

The amount of distortion as a result of the Gaussian noise can be judged from Fig. 16(a), where the embedding involving the U/V plane is least distorted, but again it is the quality of the retrieved DEM that matters. We have subjected the watermarked textures to a zero-mean Gaussian noise at standard deviations (σs) in the range 0.1 to 50. Figure 16(b) shows the graph where the mean BER is plotted as a function of σ, with both axes drawn on a

(a)

(b)

(c)

Fig. 13 Overlaying level-l approximate texture on the extracted level-l approximate depth map: (a) level 1, (b) level 2, and (c) level 3.

(a)

(b)

(c)

Fig. 14 Overlaying level-l approximate texture on the extracted level $(l-1)$ approximate depth map: (a) level 1, (b) level 2, and (c) level 3.

logarithmic scale. In general all the cases are robust until about $\sigma=0.8$, which is quite a large value. Beyond this value, robustness varies from case to case, with the fully synchronous cases being the most robust.

3.3.3 *Prospects of cropping*

The robustness demonstrated is equally valid for cropping, and we have been able to recover all the coefficients embedded in a cropped patch, provided that the following constraints were met:

Table 1 Nomenclature.

Name	synY	synCr	desynY	desynCr
Embedding plane	Y	CrCb	Y	CrCb
$L-L'$	0	0	1	1
Excluded subbands	Nil	Nil	LH_1, HL_1, HH_1	LH_1, HL_1, HH_1

Table 2 Extreme values of α for various embedding scenarios.

Case	synY	synCr	desynY	desynCr
α_{min}	12	32	4	10
α_{max}	21	49	8	13
Mean PSNR (coded texture)	18.25 dB	12.90 dB	32.94 dB	30.23 dB

1. The cropped patch has dimensions that are multiples of PD, where:

$$PD = \left(1 + 3\sum_{i=0}^{L-1} 4^i\right)^{1/2} \times t/2^{(L-L')}. \quad (4)$$

2. Each patch coordinate during the cropping must be a multiple of PD.

Calculating PD is necessary, since the embedding is done in the level-L DWT domain, and one would need the whole tree corresponding to a given coefficient. We know that each tree has as its root a coefficient from the lowest frequency subband and a set of sibling nodes. The root has three child nodes and each child, with the exception of leaves, has four child nodes. On part of the watermarked texture, all this amounts to an area of $1+3\Sigma_{i=0}^{L-1}4^i$ times the texture block size used for embedding each coefficient, which is equal to $t/2^{(L-L')} \times t/2^{(L-L')}$.

The recovery of the DWT-domain DEM has always been in its entirety, but here comes another hurdle—the classical image boundary problem—which prevents us from getting the exact inverse DWT. JPEG2000 has been supporting two kinds of transforms:[20,21] the reversible integer-to-integer Daubechies (5/3) and the irreversible real-to-real Daubechies (9/7), with the former being lossless and the latter being lossy. The standard follows a lifting approach,[21,22] for DWT in a separable rather than nonseparable manner, and approximations are inevitable at boundaries. In 2-D, the upper and lower boundaries of a given subband have to be approximated both horizontally and vertically. The module for inverse DWT usually takes into account these approximations, and the inverse must be exactly the same as the initial if the transform is supposed to be reversible. It is here that our problem starts, because even though we have recovered all the coefficients without any error, the inverse transform will still yield false coefficients at subband interfaces and borders. These errors are not affordable, as DEM is critical data. To elaborate further, let us take a visualization scenario where a number of texture images are to be tiled together for the visualization of

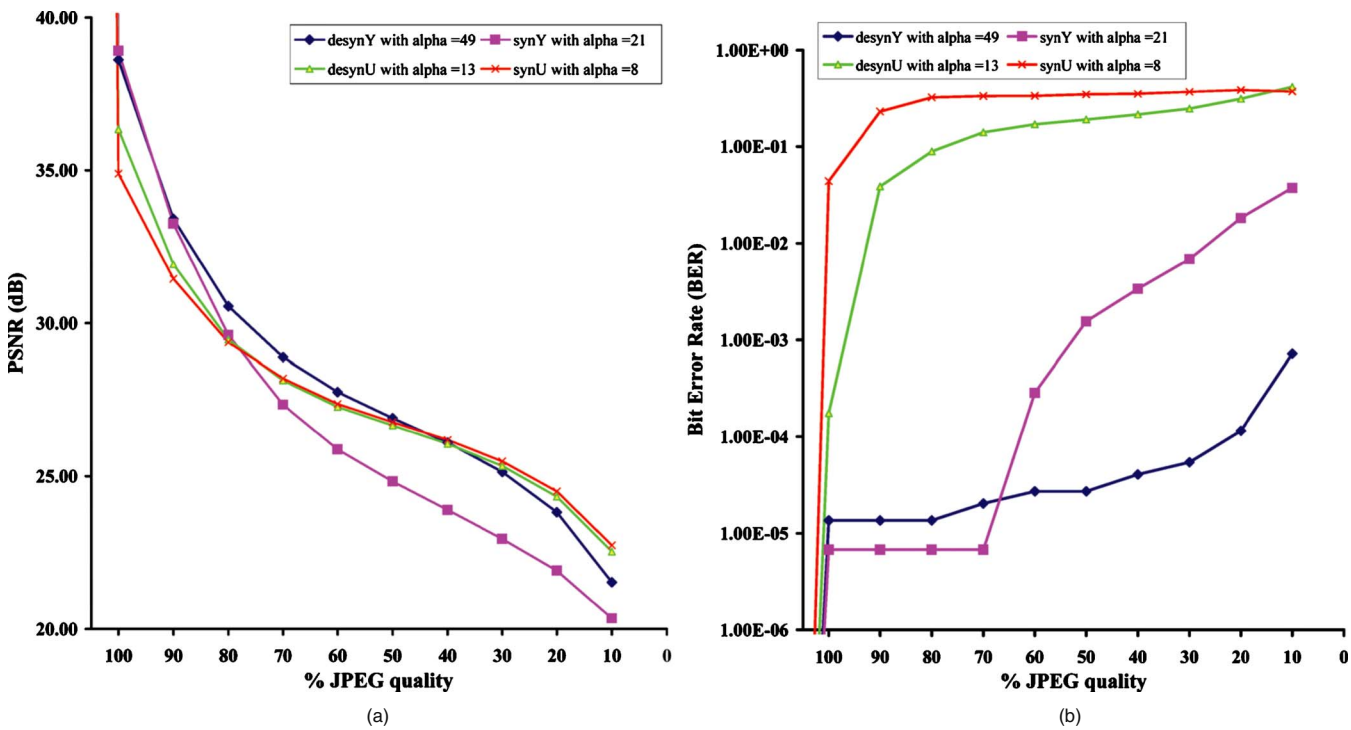

Fig. 15 Robustness of the embedded texture to JPEG compression: (a) texture quality and (b) range data quality.

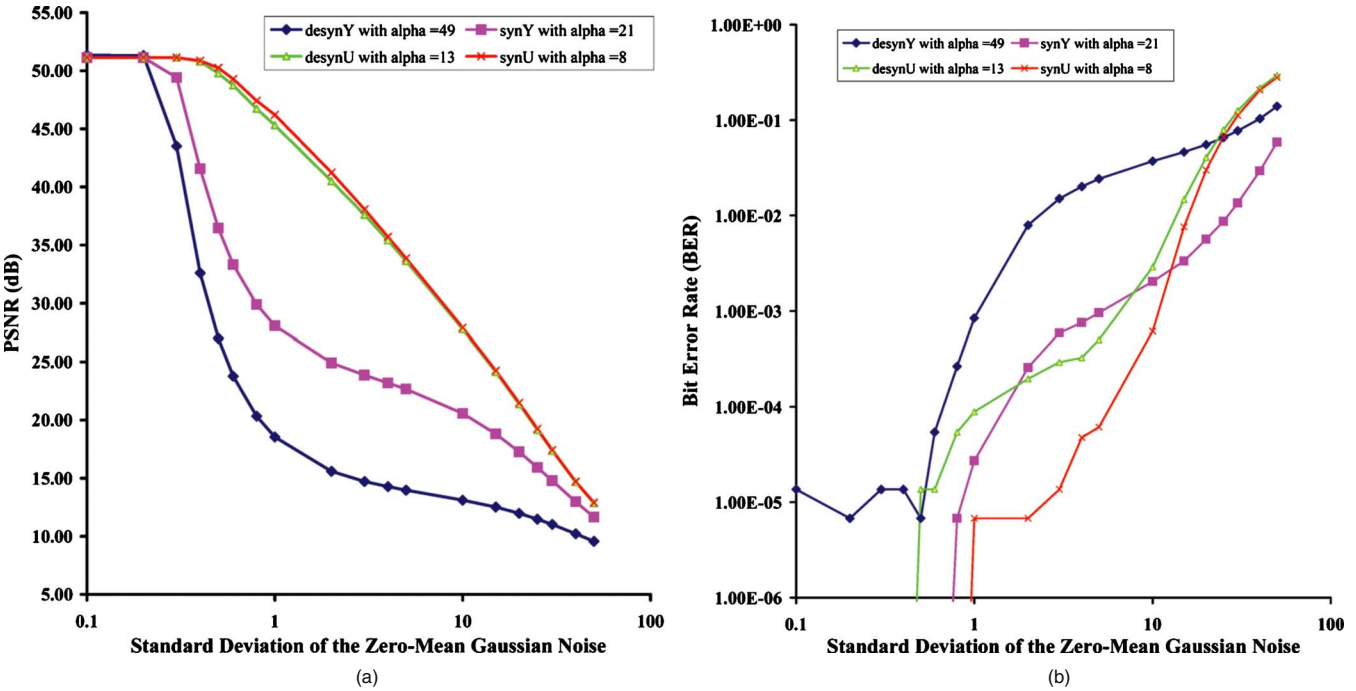

Fig. 16 Robustness to a zero-mean Gaussian noise at various standard deviations (σ): (a) texture quality and (b) range data quality.

a very large area. If we are focused at a single tile, its visualization may be rendered at full resolution but its eight neighboring tiles need not be visualized at full resolution if resources are limited. Now if suddenly the focus is changed or expanded (the observer recedes back), parts of the neighboring tiles have to be visualized. These parts will always correspond to corners of one or more neighboring tiles, and it is here our method can be helpful, since with our approach corners can more reliably be visualized via cropping. Figure 17 graphically illustrates different focus situations, with the circle representing the focus and the rectangle with broken borders representing the minimum area that must be rendered for visualization.

Let us suppose the Grand Canyon example is to be rendered at the bottom-right tile in the situation given in Fig. 17(a). Let the needed upper left corner corresponds to an area less than 256×256 pixels. Let the texture embedded synchronously at $L=2$, which would mean in light of the discussion in the beginning of this section, that after extraction from a 256×256 pixel texture patch and then inverse DWT, the DEM patch would be accurate but for the last three rows and columns of the coefficients. Figure 18 compares the 3-D visualization obtained by decoding and extracting the cropped patch by our method with that obtained from the original DEM corresponding to the patch.

4 Conclusion

In this work we present an efficient adaptive method to hide 3-D data in a texture file to have a 3-D visualization and to transmit data in a standardized way, defined by the open Geospatial Consortium.[23] The proposed method has the peculiarity of being robust and imperceptible, simultaneously. The high strength of embedding goes somewhat against security, but that has never been the goal. With adaptation in synchronization, higher quality of the depth map is ensured, but the extent of adaptation is strictly dependent on the embedding factor. The more the range data size ap-

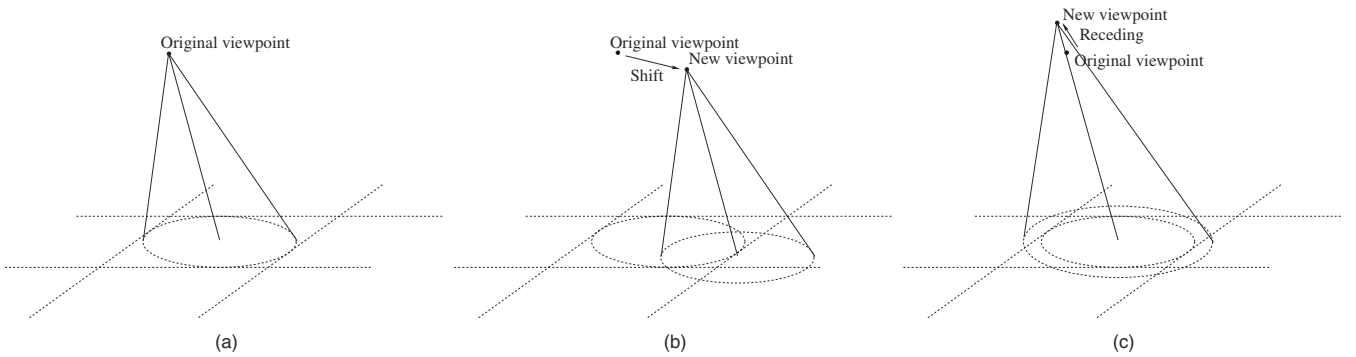

Fig. 17 Changing field of view: (a) focus center, (b) shift focus, and (c) viewer recedes.

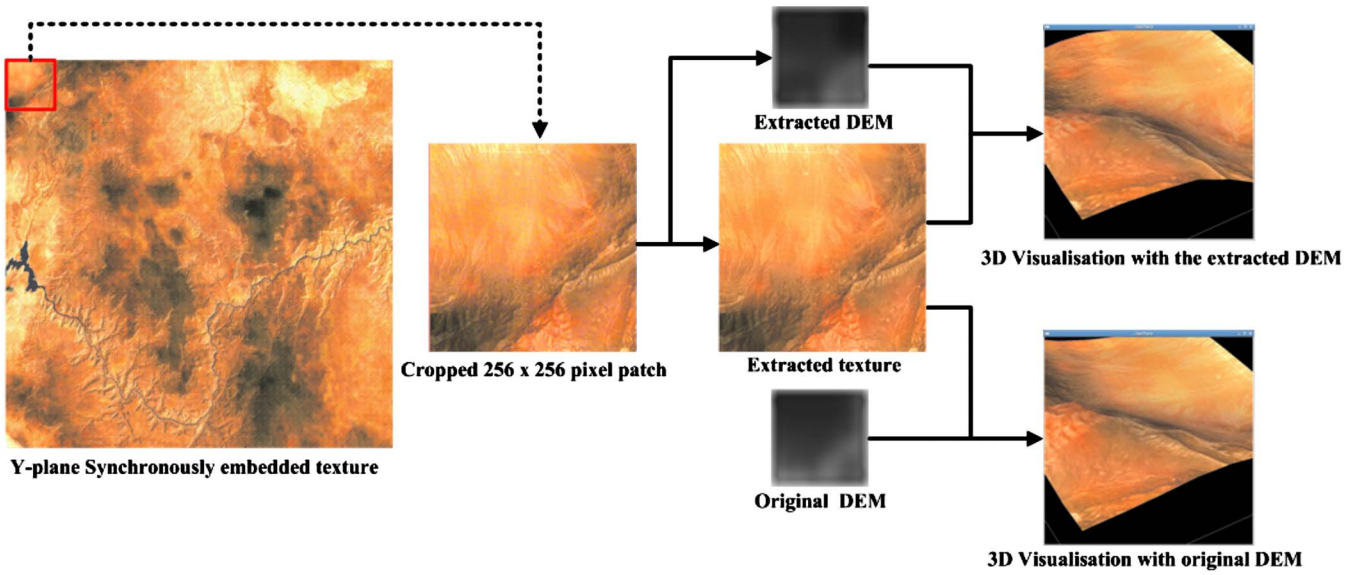

Fig. 18 3-D visualization from a patch cropped from a Y-embedded texture.

proaches that of the texture, one is compelled to elevate the strength of embedding, and if the trend goes, the prospect of using SS embedding may gradually diminish before involving more than one YCrCb planes in embedding.

The quality offered by the method for various approximations is the ultimate, as 100% of the texture and depth map coefficients are recoverable. Beyond that, one would have to look forward to the use of some other, more efficient special wavelet transforms in the JPEG2000 codec by engineering some plug-in. Ideally, these wavelets must offer the lowest quality gap between the various resolution levels. The robustness studies seem to be really interesting, especially in the case of cropping, where an effort is made to realize a 3-D visualization with a small patch cropped from the encoded image. One of our future undertakings should underline the need to further investigate the robustness, with special reference to the cropping scenario discussed. In addition, our perspective must include the study of the focus of the viewpoint.

Acknowledgments

This work is in part supported by the Higher Education Commission (HEC), Pakistan.

References

1. K. W. Bowyer, K. Chang, and P. Flynn, "A survey of approaches and challenges in 3D and multi-modal 3D+2D face recognition," *Comput. Vis. Image Underst.* **101**(1), 1–15 (2006).
2. C. Conde and A. Serrano, "3D facial normalization with spin images and influence of range data calculation over face verification," *Proc. Conf. on Computer Vision and Pattern Recognition*, vol. 16, pp. 115–120, IEEE Computer Soc., Washington, DC (2005).
3. M. P. Gerlek, *The "GeoTIFF Box" Specification for JPEG 2000 Metadata–DRAFT version 0.0*, LizardTech, Inc., Seattle (April 2004).
4. R. Lake, D. Burggraf, M. Kyle, and S. Forde, *GML in JPEG 2000 for Geographic Imagery (GMLJP2) Implementation Specification*, Number OGC 05-047r2, Open Geospatial Consortium (OGC) (2005).
5. See http://www.remotesensing.org/geotiff/spec/contents.html.
6. S. Weik, J. Wingbermuhle, and W. Niem, "Automatic creation of flexible antropomorphic models for 3D videoconferencing," in *Proc. Computer Graphics Intl. (CGI'98)*, pp. 520–527, IEEE Computer Soc., Washington, DC (1998).
7. See http://www.tixeo.com.
8. D. Q. Dai and H. Yan, "Wavelets and face recognition," in *Face Recognition*, I-Tech Education Publishing, Vienna, Austria (2007).
9. K. Hayat, W. Puech, and G. Gesquière, "Scalable 3D visualization through reversible JPEG2000-based blind data hiding," *IEEE Trans. Multimedia* **10**(7), 1261–1276 (2008).
10. J. Royan, P. Gioia, R. Cavagna, and C. Bouville, "Network-based visualization of 3D landscapes and city models," *IEEE Comput. Graphics Appl.* **27**(6), 70–79 (2007).
11. P. Gioia, O. Aubaut, and C. Bouville, "Real-time reconstruction of wavelet encoded meshes for view-dependent transmission and visualization," *IEEE Trans. Circuits Syst. Video Technol.* **14**(7), 1009–1020 (2004).
12. J. K. Kim and J. B. Ra, "A real-time terrain visualization algorithm using wavelet-based compression," *Visual Comput.* **20**(2–3), 67–85 (2004).
13. I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann Publishers, New York (2002).
14. U. Vepakomma, B. St. Onge, and D. Kneeshaw, "Spatially explicit characterization of boreal forest gap dynamics using multi-temporal lidar data," *Remote Sens. Environ.* **112**(5), 2326–2340 (2008).
15. G. Smith and D. A. Atchison, *The Eye and Visual Optical Instruments*, Cambridge University Press, New York, USA, 1997.
16. W. Puech, A. G. Bors, I. Pitas, and J. M. Chassery, "Projection distortion analysis for flattened image mosaicing from straight uniform generalized cylinders," *Pattern Recogn.* **34**(8), 1657–1670 (2001).
17. W. S. Moore, *The Basic Practice of Statistics*, W. H. Freeman Co., New York (2006).
18. See http://www.frav.es/databases/FRAV3d/.
19. See http://www.ign.fr.
20. I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Fourier Anal. Appl.* **4**(3), 247–269 (1998).
21. W. Sweldens, "The lifting scheme: a new philosophy in biorthogonal wavelet constructions," *Proc. SPIE* **2569**, 68–79 (1995).
22. S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, CA (1998).
23. See http://www.opengeospatial.org (special care must be taken on standards W3D and WVS).

Khizar Hayat received the MSc (chemistry) degree from the University of Peshawar, Pakistan, in 1993. He worked as a lecturer in the Higher Education Department, Khyber Pakhtunkhwa, Pakistan, from 1995 to 2009. During this period, he was awarded a scholarship in 2001 by the Government of Pakistan to persue an MS degree in computer science from Muhammad Ali Jinnah University, Karachi, Pakistan. He also received the Master 2 by Research (M2R) degree in computer science from the University of Montpellier II (UM2), France. In June 2009, he did his PhD while working at LIRMM (UM2) under the supervision of William Puech (LIRMM) and Gilles Gesquière (LSIS, University of Aix-Marseille) with a scholarship from The Higher Education Commission of Pakistan. He has recently joined COMSATS Institute of Information Technology (CIIT) Abbot-

tabad, Pakistan, as an assistant professor. His areas of interest are image processing and information hiding.

William Puech received the diploma of electrical engineering from the University of Montpellier, France, in 1991, and the PhD degree in signal image speech from the Polytechnic National Institute of Grenoble, France, in 1997. He started his research activities in image processing and computer vision. He served as a visiting research associate to the University of Thessaloniki, Greece. From 1997 to 2000, he was an assistant professor in the University of Toulon, France, with research interests including methods of active contours applied to medical image sequences. Between 2000 and 2008, he was an associate professor, and since 2009, he has been a full professor in image processing at the University of Montpellier, France. He works in the Laboratory of Computer Science, Robotics, and Microelectronics of Montpellier (LIRMM). His current interests are in the areas of protection of visual data (image, video, and 3-D objects) for safe transfer by combining watermarking, data hiding, compression, and cryptography. He has applications for medical images, cultural heritage, and video surveillance. He is the head of the Image and Interaction team and he has published more than ten journal papers, four book chapters, and more than 60 conference papers. He is a reviewer for more than 15 journals and for more than ten conferences. He is an IEEE and SPIE member.

Gilles Gesquière is currently an assistant professor at the LSIS Laboratory, Aix-Marseille University, France. He obtained his PhD in computer science at the University of Burgundy in 2000. His research interests include geometric modeling, 3-D visualization, and deformation. He is currently working on projects focused on 3-D geographical information systems.