

Scalable 3D Terrain Visualization through Reversible JPEG2000-Based Blind Data Hiding

K. Hayat^a, W. Puech^a and G. Gesquière^b

Abstract

In this paper a new method is presented for 3D terrain visualization via reversible JPEG2000-based blind data hiding with special focus on data synchronization and scalability. Online real-time 3D terrain visualization involves considerable amount of data. The process is essentially the mapping of the aerial photograph, called texture, onto its corresponding digital elevation model (DEM) implying at least two distinct data inputs. The presence of large disparate data necessitates a compression strategy on one hand and the integration of the DEM and texture into one unit on the other. Whilst the compression must accommodate the scalability requirement originated by the diversity of clients, the unification of data ought to be synchronous. For scalability this paper relies on the multi-resolution nature of the DWT-based JPEG2000 standard whereas the synchronized unification of DEM with the texture is realized by the application of a perceptually transparent data hiding strategy in the DWT domain. The proposed method is blind in the sense that only a secret key, if any, and the size of the original DEM are needed to extract the data from the texture image. We believe that this is one of the pioneering methods to propose scalable embedding of DEM in the texture image. The method is cost effective, in terms of memory and bandwidths, which is an advantage, especially, in real-time environments when quicker transfer of data is required. The results of a 3D visualization simulation effected with our method were encouraging and gave a useful insight to the effectiveness of our method in various network conditions.

Index Terms

3D visualization, data-hiding, Discrete Wavelet Transform (DWT), JPEG2000, Digital Elevation Model (DEM), Geographic Information System (GIS), Data synchronization.

I. INTRODUCTION

The advent of geo-browsers like Google Earth¹, World Wind² and Virtual Earth³ has brought the terrain of earth, and even its neighboring planets in the case of World Wind, to one's desktop. With each passing day the resolution of the aerial

^aLaboratory LIRMM, UMR CNRS 5506, University of Montpellier II, 161, rue Ada, 34392 MONTPELLIER Cedex 05, FRANCE ^bLSIS, UMR CNRS 6168, University of Aix-Marseille, IUT, rue R. Follereau, 13200 ARLES, FRANCE khizar.hayat@lirimm.fr, william.puech@lirimm.fr, gilles.gesquiere@lsis.org

¹<http://earth.google.com/>

²<http://worldwind.arc.nasa.gov>

³<http://www.microsoft.com/virtualearth/default.aspx>

photographs and satellite imagery which is being made available to these browsers ameliorates. All this comes at a cost as far as memory requirements are concerned. For example, a 5 meters/pixel resolution aerial photograph of a small district requires well over ten gigabytes of storage, even in the JPEG2000 compressed form. Alongside the storage is the network factor, when it comes to online real-time 3D visualization, since the data must be transferred from the server towards a client application in a timely way. In such a client/server environment the visualization quality becomes dependent on the payload, bandwidth, distant server and data transfer rate. To cope with the aforementioned challenges, any strategy that would not focus on data compression is unthinkable. But there is more to compression strategy than meets the eye since the clients are diverse in terms of resource capacity and requirements. Thus one client may have a powerful workstation while others may have just a smart-phone. Similarly one client may have a 100 Mbps connection whereas the other may be using a 56 kbps dial up modem. Still one client may want to view a precise location at high accuracy but others may just want to quickly browse through the globe. There is also a possibility that the network path of a client may involve at least one weak link or node. In between these extremes, there may be thousands of clients with varying capabilities and needs. Moreover, the preferences of a client may adapt to his capacity, *e.g.* a client handicapped by bandwidth may be more interested in real-time visualization without interruption than in high resolution. A synchronized mechanism is needed so that the server is in a better position to cater for each of its client. The wide spectrum of clients thus necessitates the data compression to be ideally scalable for each hierarchy of clients. The JPEG2000 standard [10] serves this purpose since it is wavelet-oriented and wavelets have the property of multi-resolution. The diversity of resolutions, that are resulting from a wavelet decomposition at various levels, is capable of offering the scalability needed in the situations discussed.

The 3D visualization information is in general stored in three different files: digital elevation model (DEM), an orthorectified photo and projection system and geo-referencing employed. The process of visualization in three dimensions is, essentially, the linking of the aerial photograph of the terrain, called texture, with the geometry of the terrain, called DEM. Each of the DEM coefficient represents the altitude of a particular square block of texture pixels and the process of visualization is the overlaying of texture over a regular triangle network [17], [23] obtained from the DEM. This linking is possible by geo-referencing the coordinates (longitude/latitude) of these elements depending on the scale and system of projection used. Rather than using two disparate DEM and texture files it is advisable to interleave them into a single file. For this purpose one can take a cue from the technique of data hiding and embed the smaller DEM data inside the larger texture image. The underlying assumption of our method is that the texture image must have at least 16 times more pixels than the total number of coefficients in the corresponding DEM.

It is to be noted that the context of this data hiding would be unconventional unlike steganography or watermarking as neither we are exchanging secret messages nor is copyrighting our goal. The embedding scheme must address the issue of correspondence between the texture and its DEM. In other words the integration of texture and its DEM into a single file must

be synchronous. As far as JPEG2000 is concerned there are solutions like GeoJP2 [6] and GMLJP2 [15] but these serve the purpose partially since the data is not synchronized and there is an increase in the original size of the JPEG2000 file. It would be expedient to follow a different course, to have the advantage of synchronization without any change in the JPEG2000 file size and form, by applying a scalable data hiding algorithm. This approach would neither be XML-based, like GMLJP2, nor would it introduce some kind of Universally Unique Identifier (UUID) boxes, like GeoJP2.

This work is a continuation of our previous efforts [8], [7] for the synchronized unification of the DEM and texture wherein lossless DWT decomposition level of both the message (transformed DEM) and the carrier (transformed Y plane of the texture) are the same before embedding. Moreover, in the current paper, the method is more detailed and the results have been improved. The crux of our argument is to realize a 3D visualization system that will cater for the two desirable requirements, i.e. scalability and synchronized unification of 3D data, without developing any new proprietary file format and maintaining compression performances. For scalability the DWT-based JPEG2000 standard can be relied upon for its multi-resolution characteristic due to wavelets. In addition JPEG2000 is the standard that is fast establishing itself. The synchronized integration of the disparate 3D data into one whole is achievable through the application of a perceptually transparent data hiding technique in the transform domain. All this crystallizes to the proposal of an LSB-based strategy that would embed the lossless transformed DEM in the corresponding transformed luminance component of texture of the lossless JPEG2000 coding pipeline. For the extraction step, the proposed method is blind and only the size of the original DEM is needed for extraction.

The rest of the paper is arranged as follows. Section II presents the essential concepts and previous work relevant to our approach. Our method is explained in Section III while the results which we obtained are elaborated in Section IV. Section V concludes the paper.

II. 3D TERRAIN VISUALIZATION, JPEG2000 AND DATA HIDING

This work relies on a JPEG2000-based data hiding technique in order to achieve synchronous and scalable 3D terrain visualization. In this context it would be worthwhile to present the notions of 3D terrain visualization (Section II-A) and JPEG2000 (Section II-B). Alongside, related work on data hiding in the wavelet domain is highlighted in Section II-C whereas Section II-D is dedicated to a description on the integration of geographical data into a JPEG2000 format file.

A. 3D Terrain Visualization

For years the geospatial professionals thought of mapmaking as inventing ways to flatten Earth's three-dimensional (3D) surface onto a 2D map. The various methods and projection systems which were developed are still far from reality. With the advent of 3D computer graphics it became easier to create 3D computer models to view different 3D objects from various angles as well as rotate, zoom in and out, fly through and manipulate them. However, the chronic challenge persisted because

the viewer just observed a snapshot of the 3D models projected to a 2D computer screen. The recent years' technological developments have now made it possible to create realistic 3D terrain models from scanned contour maps and other data sources⁴. The popularity and importance of 3D terrain visualization is increasing day by day in many areas, for example decision making, 3D urban models, flight and driving simulations, and more generally, multi-player applications.

Geographic data can be broadly classified into four fundamental classes: raster images, vector data, elevations and spatial databases. Images include aerial photos, scanned maps, satellite imagery and the like. Aerial photographs are the starting point for many mapping projects. By vector data is meant the traditional geometric structures like polylines, points and polygons that have been used to represent roads, streams, power lines, lakes, county boundaries and topographic lines, on a particular scale. In digital form (*e.g.* digital line graphs) this representation has the advantage of having files much less heavier than images. Moreover such a data can be resized without any loss of clarity. Elevations are usually in the form of grid of terrain heights recorded at regularly spaced horizontal intervals: the result being a DEM. DEMs are used to create 3D images of a landscape when overlaid by the corresponding 2D aerial photos or satellite images thus producing, for example, lighting effects to reveal hills and valleys. The repositories of information like census data, resource inventory data and spatial event data constitute spatial databases that are helpful to a mapmaker in special cases, *e.g.* highlighting regions in the map or forecasting changes in the terrain.

Essentially, 3D terrain representation implies to use two kinds of data combined for the visualization [19]:

- Height field data which corresponds to the elevation of terrain (Fig. 1.a). It is defined by a grid of points separated by a given step. These points are used to generate the geometry and are connected by triangles (Fig. 1.b).
- Aerial photographs can be acquired in various spectral domains such as visible or infrared. These are used to map 3D triangles previously computed (Fig. 1.c).

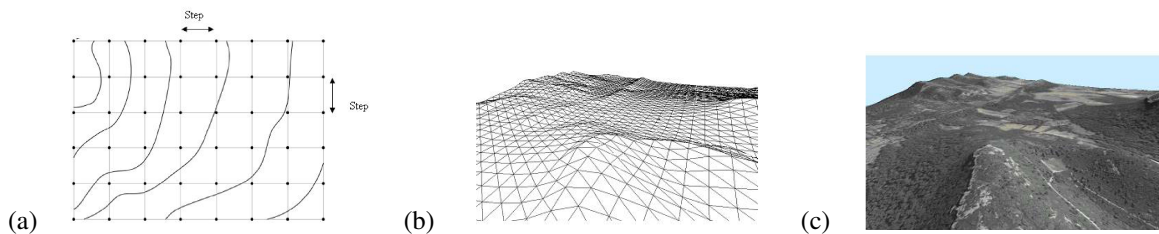


Fig. 1: a) Visualization of the uniform grid corresponding to the elevations of the terrain, b) 3D triangulated surface linking the elevations, c) Texture mapping the ortho-photograph onto the geometry.

In order to get the DEM, the simplest solution consists of using a uniform discretization of the terrain. This gives a good accuracy but leads to a quite large number of triangles, thus implying a considerable memory cost for the 3D visualization

⁴<http://www.geoplace.com>

of a large area. Many methods have been proposed to reduce the number of triangles provided by the uniform discretization, while preserving a good approximation of the original surface. One of the main approaches consists of obtaining an irregular set of triangles: Triangulated Irregular Network (TIN). A great deal of work has been done in order to create the TIN starting from a height field by using, for example, a Delaunay triangulation [5] or a progressive mesh representation [9]. Hierarchical representations of these triangulations have been proposed, thus making it possible to introduce the concept of level of detail [4]. Consequently one can obtain various levels of surfaces with an accuracy which is similar to a uniform grid but with a lower number of triangles.

To obtain a three dimensional visualization, one has to bind the texture to the terrain DEM. This bond is allowed by geo-referencing coordinates (longitude and latitude) depending on the projection system used. The information is usually stored in three different files; one for the DEM, one for the texture and one for the geo-referenced coordinates. By mapping the satellite or aerial image on the corresponding 3D terrain model. Thus, a realistic 3D model can be created with the elevations and the surface texture.

In our work we have relied on the data provided by IGN⁵ France which consists of vectorial data in the form carto[®], raster images in the form BD Ortho[®] or BD Scan[®] and DEM usually at 50 *meters* resolution. We seek to put in place the correspondence between the texture and its DEM. Using a uniform grid would allow this mapping since one would be able to do the similar thing for the ground and the images. The proposed method is about the interleaving of the above data to get a single file.

B. Discrete Wavelet Transform and JPEG2000

In standard image compression techniques, one of the essential steps is the domain transformation. The transform results in the decorrelation of the pixels. Pixel energy is thus compacted into a small number of coefficients. The idea is to enable the quantization step to trim the samples selectively, i.e. irrelevant samples must be quantized more heavily than the relevant ones [28]. One aspect of the success of JPEG [24] is in the efficiency of its transform, i.e. discrete cosine transformation (DCT) applied to each image block of 8×8 . But the use of block-based transforms rests on the unrealistic assumption of the independence of blocks.

Of the alternatives available to the block-based transforms the one that got the approval of JPEG2000 [10] proponents is the discrete wavelet transform (DWT). The key being its multi-resolution nature resulting in subbands containing some level of detail derived from the whole image or at least a considerably large tile, if the image is of very large size. The quality in such a situation is incremental as the lowest subband has the most important and relevant information and the higher subbands have finer details. Most of the energy is thus compacted into a few large transform coefficients – an entropy coder easily locates

⁵<http://www.ign.fr/>

these coefficients and encodes them. DWT offers better energy compaction than the DCT without any blocking artifact after coding. In addition the DWT decomposes the image into an L-level dyadic wavelet pyramid. The resultant wavelet coefficient can be easily scaled in resolution as one can discard the wavelet coefficients at levels finer to a given threshold and thus reconstruct an image with less detail.



Fig. 2: DWT; a) Subband structure, b) Lena, c) Lena at level 2 DWT decomposition.

The multi-resolution nature of DWT, therefore, makes it ideal for scalable image coding. After the color transformation, the DWT decomposes each component (luminance Y and chrominance Cr and Cb) into numerous frequency bands called subbands. For each level, DWT is applied twice, once row-wise and once column-wise and hence four subbands result: 1) horizontally and vertically lowpass (LL), 2) horizontally lowpass and vertically highpass (LH), 3) horizontally highpass and vertically lowpass (HL) and 4) horizontally and vertically highpass (HH). Let us consider the input image signal (or tile-component signal if image is large) as the LL_0 band. A $(R - 1)$ -level wavelet decomposition is associated with R resolution levels. Each subband of the decomposition is identified by its orientation (i.e. LL , LH , HL , and HH) and its corresponding decomposition level $(0, 1, \dots, R - 1)$. At each resolution level (except the lowest) the LL band is further decomposed. Thus the LL_0 band is decomposed to yield the LL_1 , LH_1 , HL_1 and HH_1 bands. Then, at the next level, as illustrated in Fig. 2.a, the LL_1 band is decomposed. This process is repeated until the LL_{R-1} band is obtained. If no transform is applied ($R = 1$) then there is only one subband: the LL_0 subband. A level-2 wavelet decomposition of the image Lena, given in Fig. 2.b), is illustrated in Fig. 2.c.

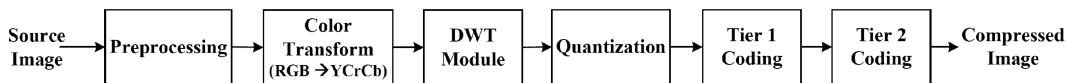


Fig. 3: A generalized scheme of the JPEG2000 encoder.

Right from the beginning, JPEG2000 has been supporting two kinds of transforms: the reversible integer-to-integer Daubechies (5/3) and the irreversible real-to-real Daubechies (9/7) [3], [27] with the former being lossless and the latter

being lossy. The 2D transformation can then be carried out by separately applying the 1D version horizontally and vertically one after the other. Let the 1D (pixel row or pixel column) input signal be $S_1, S_2, S_3, \dots, S_n$ then for the reversible 5/3 wavelet transform the lowpass subband signal $L_1, L_2, L_3, \dots, L_{n/2}$ and highpass subband signal $H_1, H_2, H_3, \dots, H_{n/2}$ are given by:

$$\begin{cases} H_i &= S_{2i+1} - \lfloor \frac{1}{2}(S_{2i+2} + S_{2i}) \rfloor \\ L_i &= S_{2i} + \lfloor \frac{1}{4}(H_i + H_{i-1}) + \frac{1}{2} \rfloor. \end{cases} \quad (1)$$

In a simplified way, a typical JPEG2000 encoder⁶ consists of the following steps [28] which are illustrated in Fig. 3:

- 1) Preprocessing such as tiling and shifting the origin of the pixel values to 0 by subtracting 128.
- 2) Intercomponent transform in the form of irreversible or reversible color transform to pass from RGB space to YCrCb space.
- 3) Intracomponent transform that may be lossy or lossless DWT.
- 4) Quantization which decreases the size of the large coefficients and nullifies the small ones.
- 5) Tier 1 coding when the quantized coefficients are partitioned into rectangular code blocks and each is subjected independently to three coding passes. This step involves entropy coding too.
- 6) Tier 2 coding which is the packetization step whereby the code pass data is converted to packets – these packets are combined to get the final image in the JPEG2000 format.

It must be noted that in a JPEG2000 coding pipeline there are two primary sources of data loss. One is obviously quantization and the other is the stage in tier-1 coding when a decision is made that which coding passes must be excluded from the final JPEG2000 file. For the method proposed in this paper, the scalability prospects offered by JPEG2000 in the form of multi-resolution is to our advantage, especially in the client/server environment.

C. Data Hiding in the Wavelet Domain

Data hiding is an established field and that is why a lot has been written about it [2], especially for the last two decades. We, therefore, focus on the literature about wavelet-based data hiding which is again very extensive and one is compelled to be brief. Data hiding deals with embedding an information, called message, inside some host signal, like image, sound or video, called cover. The message may be small and robust as in the case of copyright protection in the form of watermarking or it may be large, critical and statistically invisible as in steganography. Four factors [1] characterize the effectiveness of a data hiding method, namely hiding capacity, perceptual transparency, robustness and tamper resistance. Hiding capacity refers to the maximum payload that can be held by the cover. Perceptual transparency ensures the retention of visual quality of the cover after data embedding. Robustness is the ability of the cover to withstand various signal operations, transformations and noise

⁶<http://www.ece.uvic.ca/~mdadams/jasper/>

whereas tamper resistance means to remain intact in the face of malicious attacks. The relative importance of these four factors depends on the particular data hiding application. For example, for visually sensitive applications perceptual transparency becomes very important. Domain-wise, embedding can be carried out in the spatial domain or the transform domain. Pixel or coefficient allocation for data embedding may be regular (e.g. every k^{th} pixel) or irregularly distributed (e.g. pseudo-random). Probably the most preferred pixel allocation is by running a pseudo-random number generator (PRNG) using some secret key as a seed. Finally, an embedding method is *blind* if data extraction by the recipient does not require the original cover.

Many methods have been proposed in the literature for wavelet-based data hiding but few of these are compatible with the JPEG2000 scheme. According to [21], data hiding methods for JPEG2000 images must process the code blocks independently and that is why methods like inter-subband embedding [12] and those based on hierarchical multi-resolution relationship [14] have not been recommended. In the same breath the authors of [21] rejects the correlation-based method [29] as well as non-blind methods. The reason for this is the limited number of coefficients in a JPEG2000 code-block that are likely to fail in reliably detecting the hidden information in a single independent block. There are methods [30], [13] for embedding invisible watermarks by adding pseudo-random codes to large coefficients of the high and middle frequency bands of DWT but the methods have the disadvantage of being non-blind. The blind scheme proposed in [26] is to integrate data hiding with the Embedded Block Coding with Optimized Truncation (EBCOT) and embed data during the formation of compressed bit stream. The scheme is claimed to have robustness and good perceptual transparency. One particular technique [20] embeds watermark in the JPEG2000 pipeline after the stages of quantization and region of interest (ROI) scaling but before the entropy coding. For reliability purposes the finest resolution subbands are avoided. A window sliding approach is adopted for embedding with the lowest frequencies having higher payload. Piva *et al.* have proposed an authentication scheme that embeds an image digest in a subset of the subbands from the DWT domain [25]. The image digest is derived from the DCT of the level 1 DWT *LL* subband of the image. The resultant DCT coefficients are scaled down by quantization and ordered from most to least significant through a zig-zag scan. A most significant subset, after discarding the DC coefficient, is quadruplicated for redundancy and then rescaled and scrambled by using two different keys. This gives the message which is substituted to the subbands selected from a set obtained by the further wavelet decomposition of the level 1 *HL* and *LH* subbands of the original image. One blind method [16] transforms the original image by one-level wavelet transform and sets the three higher subbands to zero before inverse transforming it to get the modified image. The difference values between the original image and the modified image are used to ascertain the potential embedding locations of which a subset is selected pseudo-randomly for embedding. The method of Kong *et al.* embeds watermark in the weighted mean of the wavelets blocks, rather than in the individual coefficient, to make it robust and perceptually transparent [11]. While explaining their method of embedding biometric data in fingerprint images, Noore *et al.* argue against the modification of the lowest subband to avoid degradation of the reconstructed image as most of the energy is concentrated in this band [22]. Instead they propose to redundantly embed information in all

the higher frequency subbands.

D. Geographical Metadata Integration into the JPEG2000 Codestream

As far as the insertion of GIS information, as metadata, into a JPEG2000 coding pipeline is concerned there have been limited efforts to date. There are already efforts like GeoJP2 [6] and GMLJP2 [15] for the integration of geographical data with the related aerial photographs. The GeoJP2 is a GeoTIFF-inspired method for adding geospatial metadata to a JPEG2000 file. The GeoTIFF specification⁷ 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. Its aim is to allow means for tying a raster image to a known model space or map projection, and for describing those projections. GeoTIFF does not intend to become a replacement for existing geographic data interchange standards. GeoTIFF uses a small set of reserved TIFF tags to store a broad range of georeferencing information, catering to geographic as well as projected coordinate systems needs. It uses numerical codes to describe projection types, coordinate systems, datums and ellipsoids.

The additions made to the box-based JPEG2000 format are two UUID boxes, namely, the GeoTIFF box and the optional world file box. The former contains a degenerate GeoTIFF file and the resultant JPEG2000 file have the same level of geospatial metadata as is provided by the GeoTIFF standard. The mechanism is simple using the widely supported GeoTIFF implementations but the introduction of new UUID boxes have the disadvantage that there is an increase in the original JPEG2000 file size. The GMLJP2 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 meta-data, 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 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.

III. THE PROPOSED METHOD

As described in Section II-A the 3D visualization needs more than one file: the DEM and the image of texture being the essential. To coalesce this disparate data to a single JPEG2000 file, while taking into account the requirements of synchronization and scalability, we propose a data hiding-based method. An overview of the method is given in Section III-A. Section III-B presents the transformation of the DEM and the texture. The embedding is explained in Section III-C while the decoding and 3D visualization are elaborated in Section III-D.

⁷<http://www.remotesensing.org/geotiff/spec/contents.html>

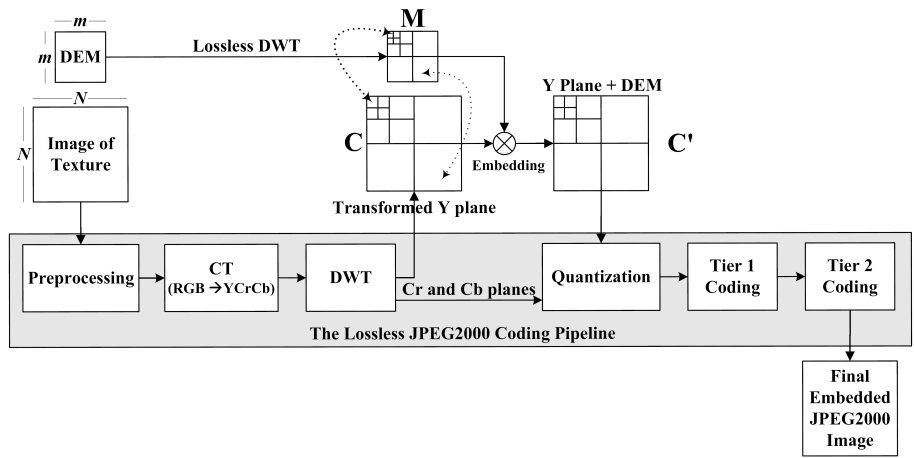


Fig. 4: Description of the method: embedding transformed DEM coefficients into the transformed coefficients of each Y texture subband.

A. Overview

In this work the thrust is on the performance of the proposed method by the inclusion/exclusion of high frequency subbands in the DWT resolution hierarchy rather than modifying the data by quantization or tier-1 coding. That is why the JPEG2000 encoder has been set at the lossless mode and the wavelet transform is integer-to-integer. The scalability prospects offered by JPEG2000 in the form of multi-resolution is to our advantage, especially in the client/server environment. In a nutshell, our method is the LSB-based embedding of lossless wavelet transformed DEM into wavelet transformed luminance (Y) plane of texture extracted from a reversible JPEG2000 coding pipeline after the DWT step. Because of the small embedding factor needed for our application, we have chosen to embed the data only in the Y component in order to have a fast extraction and to preserve a better bitrate. The marked Y plane of texture is then reinserted to the JPEG2000 coding pipeline. Two factors have been taken into account in the course of embedding. First, the level of transform in both the cases, i.e. DEM and texture, is the same. Second, there is a synchronization in embedding: low subband DEM coefficients are to be embedded into those of low subband of texture while higher ones in the higher. The method is blind since there is no need of original texture image at the decoding side. The final image format is thus JPEG2000 compliant implying that there is no need of additional technology for the proposed method. A generalized scheme of the method is given in Fig 4.

B. Transformation to get the cover and the message

The two files of DEM and texture are the inputs whereby the former would process into the message and the latter would give the cover in the data embedding step. The DEM consists of 16 bit coefficients that represent the elevations in the form of altitudes. Each of these coefficients corresponds to a particular square block of pixels in the texture. Texture is already a color image in some standard format with RGB components. From the size of the DEM we can calculate the ratio between a DEM

coefficient and its associated block of texture. Lossless DWT is applied, in isolation, to the DEM at a particular level, say L that corresponds to $R = L + 1$ resolutions. The resultant coefficients constitute the bits of our message M in the traditional sense of data hiding. In fact, to represent the DEM in the spatial domain, only 14 bits per coefficient are necessary (altitude between $-8192 m$ and $+8192 m$), but after the DWT 16 bits are necessary to represent each coefficient. Simultaneously, the RGB space texture image is introduced to a standard JPEG2000 encoder. The encoder, after necessary preprocessing, passes it from RGB space to the luminance-chrominance (YCrCb) space. In the next step, i.e. DWT step, the three YCrCb components are reversibly transformed at level L . As can be seen in Fig. 4, the encoding path is diverted, after the DWT step, as far as the luminance (Y) image is concerned. The transformed Y component is our cover C . Note that whether it is the transformation of the DEM or it is the JPEG2000 encoder for texture, the implementation of DWT is based on the lifting method [18] that employs the JPEG2000 supported reversible Daubechies $(5/3)$ [3].

C. Embedding of the Data

This section explains the embedding of the message M into the cover C to get the marked cover C' . It has been ensured that both M and C are at the same level of transformation and embedding is according to the synchronization of subbands as illustrated in Fig. 4. One important point to mention is that in this work neither any copyright problem is solved nor is there any threat to the security of message. What we want is to carry out embedding without significant loss in the visual quality of the 3D rendering. At the same time, the hiding capacity is also important since day by day the resolution quality is improving at the expense of data size. Hence, of the traditional requirements of data hiding, we are more particular about perceptual transparency and payload which implies that robustness and tamper resistance are secondary in importance.

From a $N \times N$ pixel texture image and the corresponding map of $m \times m$ altitudes, we deduce the embedding factor $E = m^2/N^2$ altitude coefficients per pixel. To ensure a spatial coherence between the altitudes and the texture, the cover C is partitioned into square blocks of $\lceil 1/E \rceil$ coefficients for data embedding and every such block would hide one transformed altitude coefficient of the message M . We thus achieve a synchronization in the embedding as far as the incremental levels of the wavelets are concerned, i.e. low resolution coefficients of the altitude map are embedded in the low resolution subbands of texture whereas high resolution coefficients of the altitude map are embedded in the high resolution subbands. In this way the transmission of the part concerned with the low resolution of the texture map enables us to directly access the corresponding low resolution part of the altitude map.

For perceptual transparency, an embedding strategy based on least significant bit (LSB) substitution is proposed. The data embedding is carried out by modifying the LSB's of a certain number of coefficients of the luminance plane of the texture. An allocation policy is needed to determine which coefficients are to carry the message. These coefficients are chosen by using a PRNG with a key, K , as a seed. The use of a secret key allows us to restrict access to DEM data. Since the chosen JPEG2000

coding is reversible the hidden message can be extracted losslessly. A high level description of the embedding strategy is outlined in Algorithm 1.

Algorithm 1 The embedding strategy

```

1: begin
2: read  $C$  which is level- $L$  transformed Y texture from the JPEG2000 coding pipeline
3: read the corresponding DEM file as an image
4: apply lossless DWT to DEM at level  $L$  to get  $M$ 
5: for each subband of  $C$  and its counterpart in  $M$  do
6:   partition the  $C$  subband into square blocks in relation to the  $M$  subband
7:   initialize a PRNG with a seed  $K$ 
8:   for each correspondence of  $C$  subband block  $B_{u,v}$  and its  $M$  subband coefficient  $d_{u,v}$  do
9:     for  $k = 0$  to  $15$  do
10:      generate the index which is the next pseudo-random number to get  $i, j$ 
11:      get the  $C$  subband coefficient  $t_{i,j}$ 
12:      substitute the LSB of  $t_{i,j}$  with the  $k^{th}$  bit of  $d_{u,v}$ 
13:    end for
14:  end for
15: end for
16: label the block-wise marked  $C$  as  $C'$ 
17: end

```

In the Fig. 5 we illustrate the low level detail of the embedding procedure. Let the $m \times m$ transformed DEM coefficients are represented by $d_{u,v}$, where $u, v \in \{0, \dots, m-1\}$. The $N \times N$ transformed Y plane of texture is consequently partitioned into m^2 blocks $B_{u,v}$, with $u, v \in \{0, \dots, m-1\}$. Each of $B_{u,v}$ is composed of r^2 ($= N^2/m^2$) coefficients $t_{i,j}$, with $i, j \in \{0, \dots, r-1\}$. Let $b_{15}b_{14}b_{13}\dots b_1b_0$ and $\beta_7\beta_6\beta_5\dots\beta_1\beta_0$ be the bitwise representations of 16 bit $d_{u,v}$ and the 8 bit $t_{i,j}$, respectively. Embedding involves the hiding of each of the 16 b_k bits in as many different coefficients $t_{i,j}$ of the given $B_{u,v}$ block. These coefficients are selected by running a PRNG using the key K for one time initialization. The PRNG that will generate 16 numbers per $B_{u,v}$ which correspond to the allocated $t_{i,j}$ coefficients. Care must be taken to avoid collisions so that the same subscript is not generated twice. One of the b_k bits would substitute the LSB of the allocated $t_{i,j}$ coefficient $\beta_7\beta_6\beta_5\dots\beta_1\beta_0$ that would yield $t'_{i,j} = \beta_7\beta_6\beta_5\dots\beta_1b_k$ after embedding. Hence 16 of the r^2 coefficients of each of the embedded block would be changed from $t_{i,j}$ to $t'_{i,j}$. The procedure mentioned in the previous lines is followed for each block

of the transformed Y texture and for each subband of C , as described in Algorithm 1.

The resultant embedded image, C' , i.e. marked Y plane of the texture in the wavelet domain, is then re-inserted into the JPEG2000 pipeline at the same point from where it was extracted. The final encoded image thus carries the DEM coefficients hidden in some DWT coefficients.

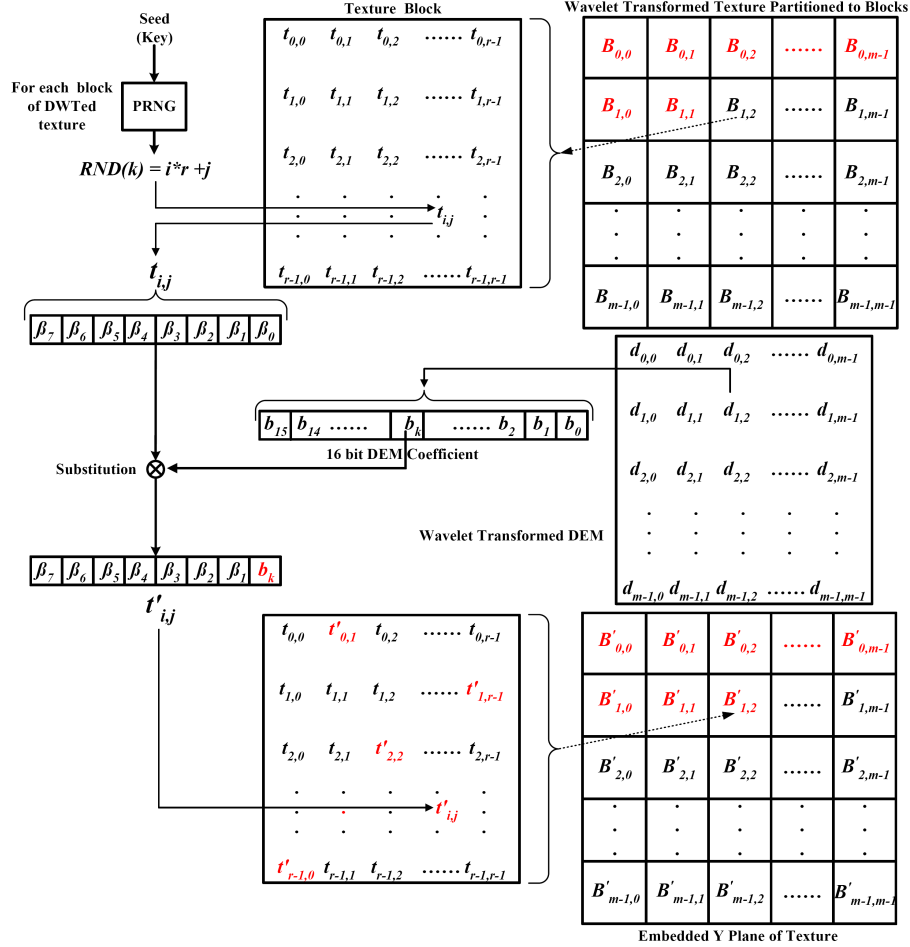


Fig. 5: Embedding a subband of the transformed DEM coefficient in a block from a subband of the transformed texture.

D. Extraction and Scalable 3D Reconstruction and Visualization

The above coded image can be utilized like any other JPEG2000 image and sent across any communication channel. The decoding is the reverse of the above process. Since the method is blind, only the secret key K is needed, along with the size of the original DEM, to recover the hidden DEM coefficients at the receiving end. Just before the inverse DWT stage of the JPEG2000 decoder, the DEM can be extracted using the above mentioned partitioning scheme and PRNG with K as the seed. All the DEM bits are LSB's of the coefficients, indexed by the PRNG sequence, in the luminance plane of the carrier texture.

One advantage of the method is in the fact that the DEM and texture can be reconstructed at the original size, even if we have only a small subset of the coefficients of the carrier. The resolution scalability of wavelets and the synchronized character

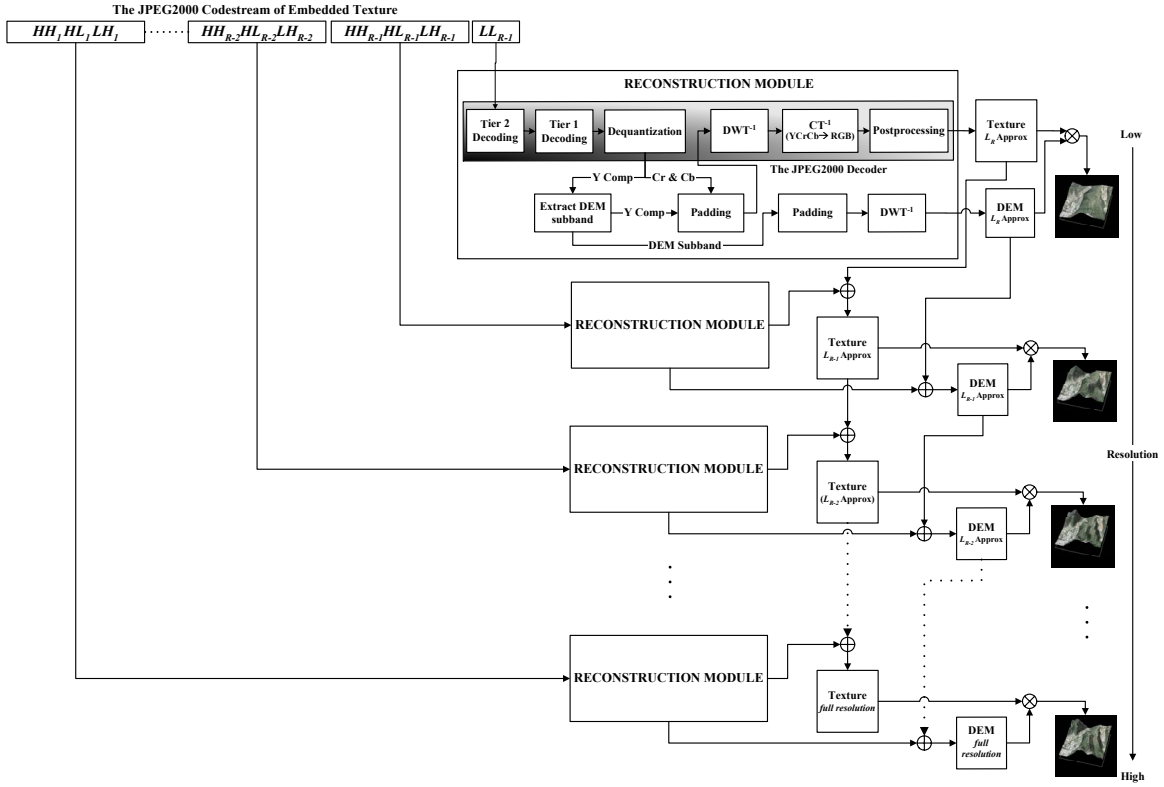


Fig. 6: Decoding, data extraction and scalable 3D reconstruction and visualization.

of our method enable a 3D visualization even with a number of layers smaller than the number of original resolution layers as a result of partial or delayed data transfer. The method thus enables to effect a visualization from a fraction of data in the form of the lowest subband, of a particular resolution level. The idea is to have a 3D visualization utilizing lower frequency $3L' + 1$ subbands out of the initial $3L + 1$ subbands ($L' \leq L$), by padding a 0 for each of the coefficient of the rest of the high frequency $L - L'$ parts. Indeed, it is always possible to stuff 0's for the higher bands. Hence a reconstruction with level L' lowest frequency subband is effected with just $(1/4^{L'}) \times 100$ percent of the count of the original coefficients: the result would be what is called *level L approximation image*. With this padding approach, irrespective of the number of subbands used for the reconstruction, the reconstructed DEM and texture have the same spatial sizes as the original ones. The reconstruction procedure from a JPEG2000 codestream of the marked texture is depicted in Fig. 6.

It is important to note that level L wavelet decomposition corresponds to $R = L + 1$ resolution levels in the range L to 0 in ascending order of quality as well as quantity of data. This implies R different visualizations of which the first one requires only the lowest LL_{R-1} subband. The second visualization would require four subbands which include the previous one, LL_{R-1} , and three others namely LH_{R-1} , HL_{R-1} and HH_{R-1} . Seven subbands, LL_{R-1} , $\bigcup_{i=1}^2 LH_{R-i}$, $\bigcup_{i=1}^2 HL_{R-i}$ and $\bigcup_{i=1}^2 HH_{R-i}$, constitute the data for third visualization. In general l^{th} visualization is based on $3(l-1) + 1$ lowest subbands

$LL_{R-1}, \bigcup_{i=1}^{l-1} LH_{R-i}, \bigcup_{i=1}^{l-1} HL_{R-i}$ and $\bigcup_{i=1}^{l-1} HH_{R-i}$. All the $3(R-1)+1$ subbands are necessary for the highest quality R^{th} visualization. In a loose sense, any given resolution is a superset of all the preceding ones. The scalability has its origin in the fact that one can have R different images of approximation. Choice can be dynamically made among these R different options depending on the client's resources, constraints and requirements as well as network and distance considerations. A client with a powerful platform and high available bandwidth may be able to download all the data timely and decode the level 0 JPEG2000 image to get the highest quality visualization. On the other hand a client on a smart-phone and weak communication signal may have to be content with a small subset of subbands (Fig. 6) which it will stuff with 0's for higher frequency bands during the decoding process to get the approximation rather than the original image with the improvement of signal he may have the luxury to go for higher quality. All this becomes more critical in the real-time case since now the frame rate, for example, can become more important than the quality and one can compromise over latter to certain extent. In essence, even for a very small fraction of the number of initial coefficients, of the texture and its DEM, one can have a 3D visualization that is considerably cost effective.

IV. RESULTS

This section has two parts. In Section IV-A we analyze the application of our method when applied to an example altitude map of 64×64 coefficients (Fig. 7.a) associated with a 3200×3200 pixel texture (Fig. 7.b). For the purpose of comparison a 128×128 pixel detail of the above mentioned image of texture is presented in Fig. 7.c. Section IV-B is concerned with the 3D visualization simulation with respect to another example that is to be demonstrated at high and low bitrates.



Fig. 7: Example images; a) Original altitude map, b) Original texture map, c) A part of texture magnified.

A. Analysis of the Method as Applied to a Real World Example

For the example illustrated in Fig. 7, each coefficient of the altitude is coded with 2 bytes implying the embedding factor of 1 coefficient per 50×50 pixels of texture. The application of a level 1 lossless wavelet transformation to the DEM results in Fig. 8.a. The corresponding transformed Y-plane of the texture (Fig. 8.b) is extracted *in situ* from the JPEG2000 encoding process of the texture image. The same set of images for level 3 are illustrated in Fig. 8c and d.

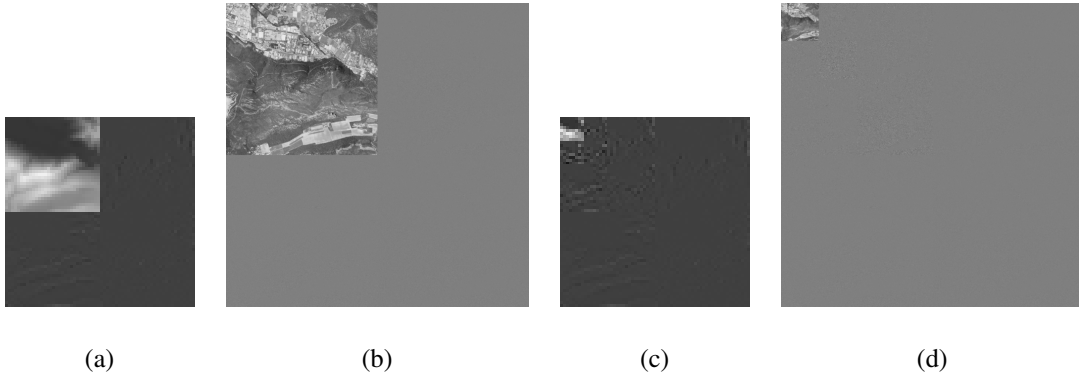


Fig. 8: DWT at Level 1: a) altitude, b) Y Plane of texture. DWT at Level 3: c) altitude, d) Y Plane of texture.

To analyze the quality of the embedded texture image, with respect to the original, the measure of PSNR (peak signal to noise ratio) has been employed:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE}, \quad (2)$$

where mean square error (MSE) is a measure used to quantify the difference between the initial image I and the distorted image I' . If the image has a size of $M \times N$ then:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I(i, j) - I'(i, j))^2. \quad (3)$$

In order to judge compression performance, a measure called bitrate, which is the amount of bits necessary per pixel, is utilized:

$$Bitrate = \frac{\text{Compressed Image Size in bits}}{\text{Original Image Size in pixels}} \text{ in bits per pixel (bpp).}$$

Strictly speaking, DEM is a set of altitude coefficients rather than pixels which requires some modifications in the definition of the above measures. Hence, instead of bitrate, a measure we call bits per coefficient (i.e. amount of information per altitude coefficient) has been used. Since altitudes are expressed in meters, it would be better to use MSE in square meter unit for error measurement. For a more comprehensive explanation one can rely on the one dimensional version of root mean square error (RMSE) which is the positive square root of MSE: $RMSE = \sqrt{MSE}$. Embedding of the transformed DEM in the transformed texture takes place according to the policy explained in Section III. It is worthwhile to note that, in general, the data embedding at level L decomposition takes place in $3L + 1$ partitions. Hence for the DWT at level 1 (Fig. 8.a) the embedding has been done in 4 partitions (LL_1 , LH_1 , HL_1 and HH_1) whereas for level 3 wavelet decomposition, the embedding of data has been realized in 10 partitions. The marked luminance plane of texture is reintroduced to the JPEG2000 coding pipeline at the point where it was extracted and thus the output is JPEG2000 compliant. No matter what is the level of decomposition L , degradation in the texture image, due to data embedding, amounts to a mean square error (MSE) of 0.01 corresponding to a PSNR of 68.5 dB. This is attributed to the meager embedding factor of 6.4×10^{-3} altitude coefficients per pixel since only one 16 bit DEM coefficient is embedded per 50×50 block of coefficients of the Y component of texture.



Fig. 9: Reconstruction of the images from the approximation image at Level 1; a) Extracted altitude, b) Difference image between the original altitude and the extracted one, c) Texture, d) Magnified part of texture corresponding to Fig. 7.c.



Fig. 10: Reconstruction of the images from the approximation image at Level 2; a) Extracted altitude, b) Difference image between the original altitude and the extracted one, c) Texture, d) Magnified part of texture corresponding to Fig. 7.c.

Fig. 9.a shows an altitude map reconstructed from the coefficients extracted from a level 1 image of approximation (LL_1) of the embedded texture image. The reconstruction has been done by stuffing 0's in place of the three higher subbands and subsequently applying inverse DWT. Thus only 25% of the count of the initial coefficients have been used for reconstruction. The difference of this image with the original altitude is illustrated in Fig. 9.b. When the above process of reconstruction is applied to the level 1 lowest subband of the embedded texture image Fig. 9.c is obtained. For comparison with the original example, the 128×128 pixel magnified part is shown in Fig. 9.d. The same sequence of four figures, i.e. *a* to *d*, is maintained in each of the Fig. 10– 12 which corresponds to the reconstruction from levels 2 to 5 lowest subbands, respectively. For subjective analysis, one can visually compare the part *c* of Fig. 7, with the same of each of Fig. 9– 12. There is certainly degradation in the visual quality but it is far more less in the face of the quantity of data utilized for reconstruction, e.g. level 3 reconstruction has been carried out with only 1.56% of the total coefficients. The degradations in the quality of texture are presented in Fig. 13 in the form of difference images in terms of the magnified part corresponding to Fig. 7.c.

Objectivity demands a quantitative comparison which we have done in terms of the images of difference by observing the measures, like MSE or PSNR, as a function of bitrate. The results are summarized in Table I for the DEM and Table II for the texture. The effectiveness of our method is revealed by the fact that even for a bitrate as low as 0.26 *bpp* (corresponds to

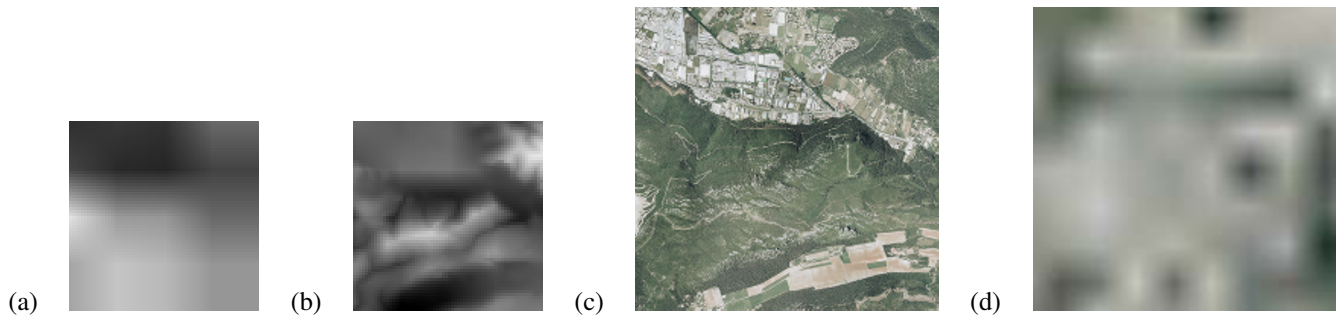


Fig. 11: Reconstruction of the images from the approximation image at Level 4; a) Extracted altitude, b) Difference image between the original altitude and the extracted one, c) Texture, d) Magnified part of texture corresponding to Fig. 7.c.

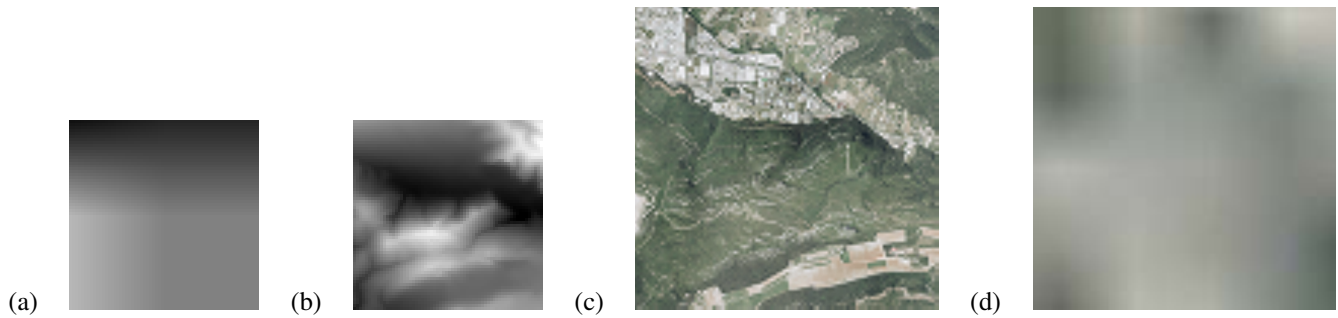


Fig. 12: Reconstruction of the images from the approximation image at Level 5; a) Extracted altitude, b) Difference image between the original altitude and the extracted one, c) Texture, d) Magnified part of texture corresponding to Fig. 7.c.

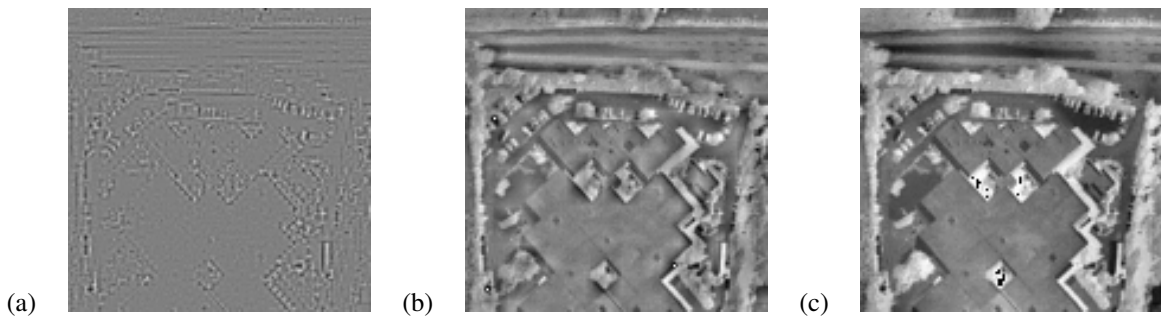


Fig. 13: Difference between the original luminance and reconstructed texture (for the magnified part of texture corresponding to Fig. 7.c) of; a) Level 1, b) Level 3, c) Level 5.

level 3 approximation image of texture, Table II) one observes a $PSNR$ of 25.47 dB corresponding to a $RMSE$ for DEM of 22.02 meters (Table I). Given the fact that being computed from a small fraction of initial data, this error is tolerable if the observation is made from a high altitude. To elaborate further, MSE has been plotted as a function of bitrate for the texture (Fig. 14.a.) and as a function of bits per coefficient for DEM (Fig. 14.b). In the latter case, we have plotted MSE against bits per coefficient due to the fact that rather than the whole JPEG2000 encoding only DWT has been applied to the DEM coefficients. As can be seen in Fig. 14.a. the MSE decreases as further levels of information are added to the image of

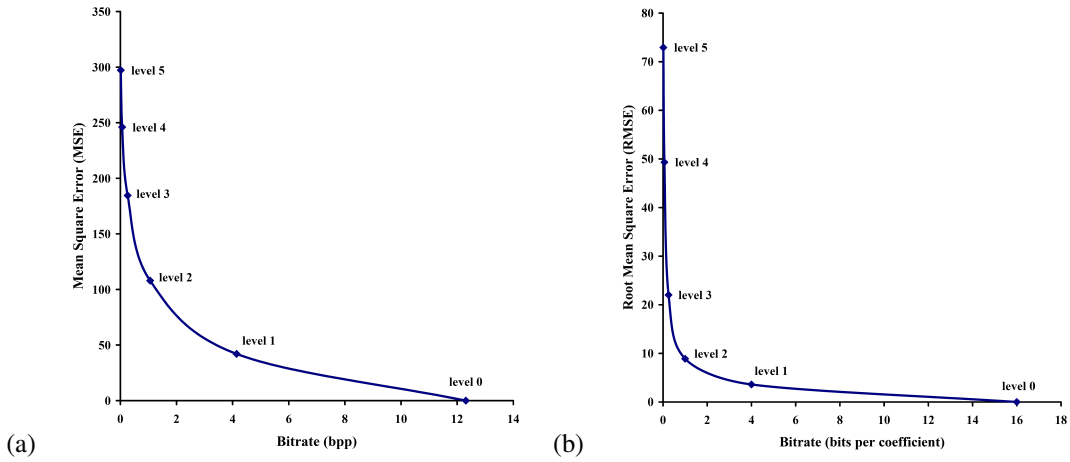


Fig. 14: Graphical display of variation in quality as a function of bitrate: a) Texture, b) DEM.

approximation, e.g. for 1.06 *bpp* (level 2) the MSE is 107.97 and for 4.14 *bpp* (level 1) the MSE is 42.01. A similar trend is observed in case of *RMSE* for the DEM (Fig. 14.b). For 3D visualization, the acceptable quality of both DEM and texture is a function of the distance between the 3D surface and the viewpoint. The more this viewpoint is near the 3D surface, better must be the quality of both DEM and texture.

Approximation image	lev. 5	lev. 4	lev. 3	lev. 2	lev. 1.	lev. 0
Bits per coefficient	0.016	0.062	0.25	1	4	16
RMSE (meter)	72.92	49.33	22.02	8.90	3.60	0
PSNR (<i>dB</i>)	10.87	14.27	21.27	29.14	37	∞

TABLE I: Results obtained for DEM after the extraction and reconstruction as a function of the used data (*for level 0, all of the transmitted data is used for reconstruction*).

Approximation image	lev. 5	lev. 4	lev. 3	lev. 2	lev. 1.	lev. 0
Compression ratio	1508.62	376.82	92.38	22.64	5.80	1.95
Bitrate (bpp)	0.016	0.064	0.26	1.06	4.14	12.31
MSE	297.22	246.04	184.56	107.97	42.01	0.01
PSNR (<i>dB</i>)	23.40	24.22	25.47	27.80	31.9	68.53

TABLE II: Results obtained after the extraction and reconstruction of the texture as a function of the used data int the form of embedded texture.

The reconstructed 3D surfaces drawn from the various approximation DEMs are shown in Fig. 15. These surfaces when overlaid by their corresponding textures effect 3D visualizations (Fig. 16). One can compare the final result between a 3D visualization with all the data (Fig. 16.a), after being subjected to our method, and a visualization with the level 1, 2, 3 and 4

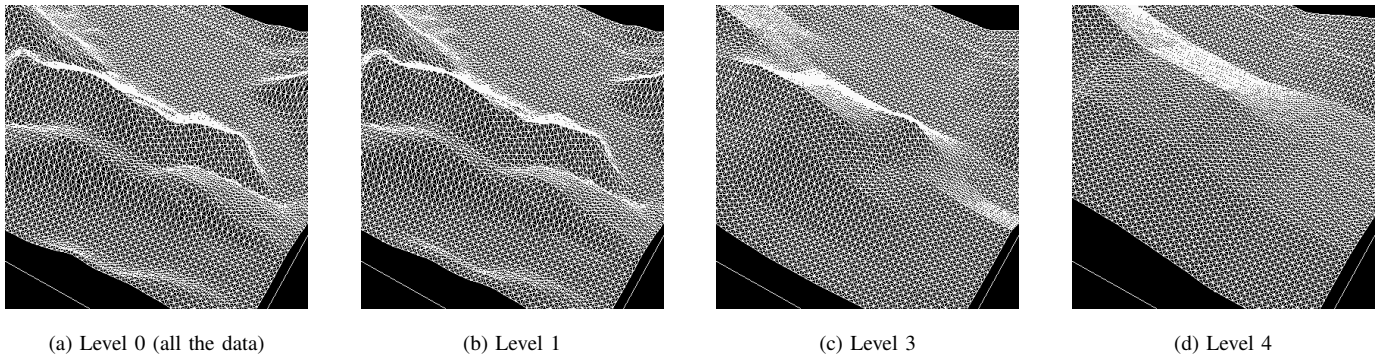


Fig. 15: 3D surface representation using the extracted DEMs and obtaining their approximations at various levels.

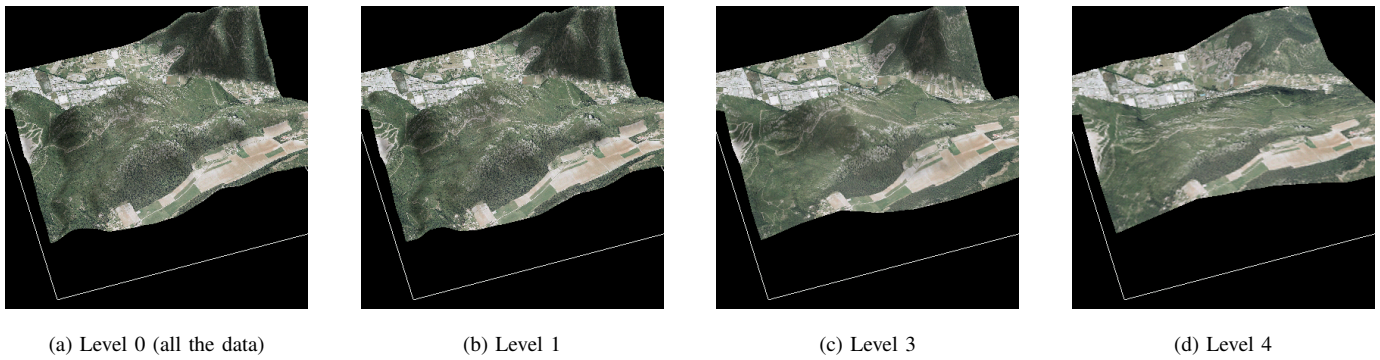


Fig. 16: 3D visualization with the images of approximations at various levels.

(Fig. 16.b–e), corresponding to 25%, 6.25%, 1.56% and 0.39% of the transmitted coefficients, respectively.

B. Simulation Example

To present an application of our method it would be worthwhile to run a practical visualization simulation. The inputs used in the simulation example are given in Fig. 17. The DEM (64×64) from Fig. 17.a and its corresponding 3200×3200 aerial image (Fig. 17.b) are subjected to our method. Two series of simulations were effected with one based on full resolution, i.e. level 0 approximation images, and the other on level 4 images of approximation. Some snapshots of the simulations, at regular intervals, from a transmitted image sequence of more than 300 images are presented.

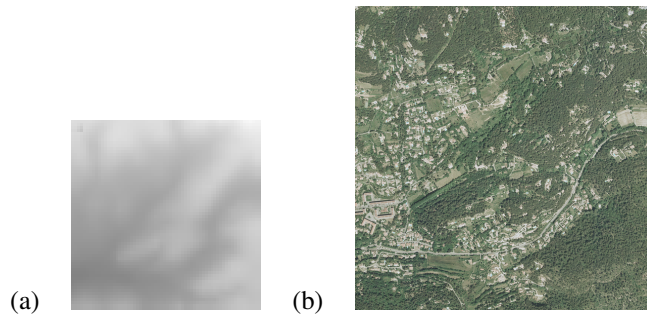


Fig. 17: Example altitude/texture pair utilized in the visualization simulation: a) original altitude map, b) original texture map.

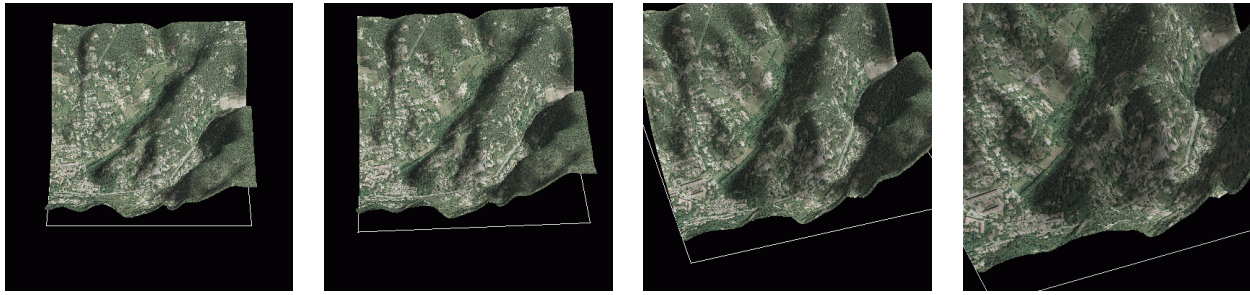


Fig. 18: Snapshots of the 3D visualization simulation based on **level 0 (for 240 Mbps)** images of approximation extracted and reconstructed from the DEM embedded texture.

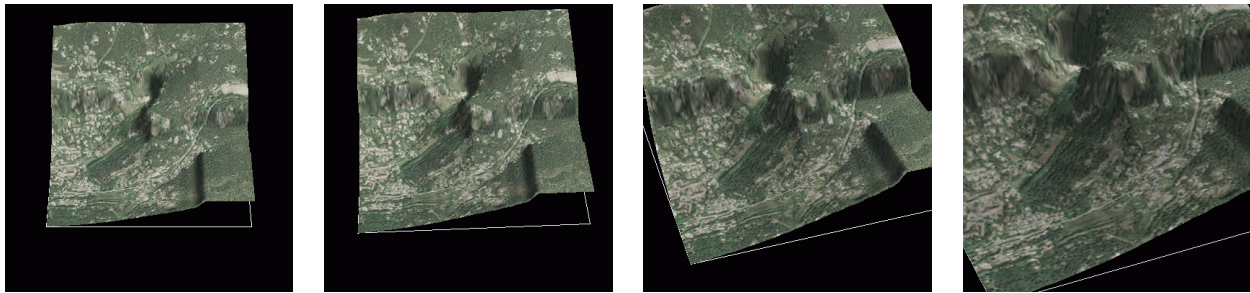


Fig. 19: Snapshots of the 3D visualization simulation based on **level 4 (for 640 kbps)** images of approximation extracted and reconstructed from the DEM embedded texture.

Five different snapshots from the 3D visualization of our example are illustrated Fig. 18 for level 0 (corresponding to a bitrate of 240 Mbps) and Fig. 19 for level 4 (corresponding to a bitrate of 640 kbps). The snapshots are at regular intervals when the distant observer is coming closer and closer to the terrain and such they are taken as a function of the decreasing aerial distance of the viewer from his point of focus. The viewer's aerial position is the same for the two bitrates (240 Mbps in Fig. 18 and 640 kbps in Fig. 19) at a given interval. The difference is obvious but not glaring given the fact that the data of Fig. 19 corresponds to only 0.39% of the number of coefficients of the data of Fig. 18. This lower than expected degradation in quality motivate us to have an effective scalable visualization. For example a 3200×3200 aerial image requires 30 MB of storage employing a 240 Mbps bandwidth requirement if only one image is to be transferred per second. In other words a 100 Mbps connection would require 2.4 seconds for a single image to transfer and the same for wireless networks is around 1200, 625, 133 and 48 seconds for EDGE (200 Kbps), 3G (384 Kbps), HSDPA (1.8 Mbps) and WIFI (5 Mbps), respectively. A level 4 approximation of the same embedded image in JPEG2000 format would have a size in the order of 80 KB implying the requirement of 640 kbps for transferring one image per second. With EDGE one can transfer one such image in about 3 seconds and WIFI can transmit 8 such images per second. Level 5 approximation may further reduce the payload and now one can transfer one image per second over EDGE and 32 images over WIFI making the latter suitable for video based streaming.

For the level 4 approximation, a closer examination of Fig. 19 reveals that of the DEM/texture pair it is the DEM which is affected the most in terms of quality. This means that the DEM is more sensitive than the texture to a partial reconstruction. This sensitivity of DEM necessitates the fact that the level of wavelet decomposition could be lower for DEM than texture before embedding. Since the size of DEM file is much smaller than that of the texture, the lowest frequency subband at some level can be used for embedding. But this would undermine the perceptual transparency since the embedding density of the energy-richer part would be increased. Moreover, the synchronization management between DEM and texture would be more difficult and this could be costlier in terms of time.

V. CONCLUSION

In this paper we presented a new method for a scalable 3D terrain visualization through reversible JPEG2000-based blind data hiding. This paper is focused on the topic of data synchronization and scalability. The results reveal that the proposed method offers at least three advantages. First is the synchronized integration of disparate 3D data into one whole by the application of data hiding. The second advantage is the scalability in 3D visualization through the utilization of JPEG2000 supported DWT. Last, but not the least, is the integrability of the method with the JPEG2000 encoders to result in a monolithic standalone JPEG2000 format file that eliminates the need to develop any additional technology or data format, thus implying portability and conformance which is asked by our industrial constraints. In addition the approach is cost effective in terms of memory and bandwidths. The results shown in the case of our examples are witness to this fact since even with a tiny number of coefficients a comparatively better 3D visualization was effected. The resolution scalability of wavelets enables this 3D visualization to improve incrementally with the reception of higher frequencies/subbands. Besides, this property is helpful in real-time environment when quicker transfer of data is required. The results of 3D visualization simulation give a useful insight to the effectiveness of our method in various network conditions.

In the continuation of this work it would be worthwhile to develop a method based on lossy wavelets to further decrease bitrate. This option will however eliminate, to a considerable extent, the possibility of very high quality visualization since the lossy case is not fully reversible. A situation may thus arise that losing details of texture becomes less important than that of DEM. Using a desynchronized algorithm would be a good way and should be taken into consideration in the near future. The LSB-based embedding strategy adopted in this work has to be reconsidered for the lossy case and it will be important to explore some other embedding strategies, like the spread spectrum embedding [2], in order to keep DEM quality high for the reconstruction. As far as triangulation is concerned, there is also every likelihood of using a non-uniform grid on various levels of details, thus allowing a considerable reduction in the number of triangles necessary for a good representation of the terrain. In this paper links between tiles are not managed but our future focus is likely to be on the strategies to generate a soft transition between several tiles without cracks. In future, the streaming management between the server and its clients is

also being mulled.

VI. ACKNOWLEDGMENT

This work was supported in part by the Higher Education Commission (HEC) of Pakistan.

REFERENCES

- [1] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for Data Hiding," *IBM Systems Journal*, vol. 35, no. 3-4, pp. 313–336, February 1996.
- [2] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. Morgan Kaufmann Publishers, 2002.
- [3] I. Daubechies and W. Sweldens, "Factoring Wavelet Transforms into Lifting Steps," *Fourier Anal. Appl.*, vol. 4, no. 3, 1998.
- [4] L. D. Floriani and E. Puppo, "Hierarchical Triangulation for Multiresolution Surface Description," *ACM Trans. Graph.*, vol. 14, no. 4, pp. 363–411, 1995.
- [5] R. J. Fowler and J. J. Little, "Automatic Extraction of Irregular Network Digital Terrain Models," in *Proc. International Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH'79)*. New York, NY, USA: ACM Press, 1979, pp. 199–207.
- [6] M. P. Gerlek, *The "GeoTIFF Box" Specification for JPEG 2000 Metadata - DRAFT version 0.0*. LizardTech, Inc. 1008 Western Ave Suite 200 Seattle, WA 98104 USA, April 2004.
- [7] K. Hayat, W. Puech, and G. Gesquière, "An Efficient Data-Hiding Method Based on Lossless JPEG2000 for a Scalable and Synchronized Visualization of 3D Terrains," in *Proc. 15th European Signal Processing Conference (EUSIPCO 07)*, Poznan, Poland, September 2007, pp. 2519–2523.
- [8] K. Hayat, W. Puech, G. Gesquière, and M. Chaumont, "Wavelet Based Data-Hiding of DEM in the Context of Real Time 3D Visualization," in *Proc. SPIE, Electronic Imaging, Visualization and Data Analysis*, vol. 6495, no. 20. San Jose, CA, USA: SPIE, IS&T, January 2007, pp. 64950N(1–10).
- [9] H. Hoppe, "Progressive Meshes," in *Proc. International Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH'96)*, New Orleans, LA, USA, 1996, pp. 99–108.
- [10] ISO/IEC, *ISO/IEC 15444-1: Information technology, JPEG2000 image coding system, Part 1: Core coding system*. ISO Central Secretariat: CH-1211 Geneva, Switzerland, 2004.
- [11] X. Kong, Y. Liu, H. Liu, and D. Yang, "Object Watermarks for Digital Images and Video," *Image and Vision Computing*, vol. 22, pp. 583–595, 2004.
- [12] D. Kundur, "Improved Digital Watermarking Through Diversity and Attack Characterization," in *Proc. ACM Workshop on Multimedia Security'99*, Orlando, FL, USA, Oct. 1999, pp. 53–58.
- [13] D. Kundur and D. Hatzinakos, "A Robust Digital Image Watermarking Scheme Using the Wavelet-Based Fusion," in *Proc. IEEE International Conference on Image Processing (IEEE ICIP 97)*, vol. 1, Santa Barbara, CA, USA, Oct. 1997, pp. 544–547.
- [14] ———, "Digital Watermarking Using Multiresolution Wavelet Decomposition," in *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing (IEEE ICASP 98)*, vol. 5, Seattle, WA, USA., May 1998, pp. 2969–2972.
- [15] R. Lake, D. Burggraf, M. Kyle, and S. Forde, *GML in JPEG 2000 for Geographic Imagery (GMLJP2) Implementation Specification*. Open Geospatial Consortium (OGC), 2005, no. OGC 05-047r2.
- [16] J. L. Liu, D. C. Lou, M. C. Chang, and H. K. Tso, "A Robust Watermarking Scheme Using Self-Reference Image," *Computer Standards & Interfaces*, vol. 28, pp. 356–367, 2006.
- [17] F. Losasso and H. Hoppe, "Geometry Clipmaps: Terrain Rendering Using Nested Regular Grids," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 769–776, 2004.
- [18] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1998.
- [19] A. Martin, G. Gesquière, W. Puech, and S. Thon, "Real Time 3D Visualisation of DEM Combined with a Robust DCT Based Data-Hiding Method," in *Proc. SPIE, Electronic Imaging, Visualization and Data Analysis*, vol. 6060. San Jose, CA, USA: SPIE, IS&T, January 2006, pp. 60600G(1–8).
- [20] P. Meerwald, "Quantization watermarking in the JPEG2000 coding pipeline," in *Proc. Communications and Multimedia Security Issues of The New Century, IFIP TC6/TC11 Fifth Joint Working Conference on Communications and Multimedia Security, CMS '01*, R. Steinmetz, J. Dittmann, and M. Steinebach, Eds. Kluwer Academic Publishing, May 2001, pp. 69–79.
- [21] P. Meerwald and A. Uhl, "A Survey of Wavelet-Domain Watermarking Algorithms," in *Proc. SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents III*, vol. 4314. San Jose, CA, USA: SPIE, IS&T, January 2001, pp. 505–516.

- [22] A. Noore, R. Singh, M. Vatsa, and M. M. Houck, "Enhancing Security of Fingerprints through Contextual Biometric Watermarking," *Forensic Science International*, vol. 169, no. 2–3, pp. 188–194, 2007.
- [23] R. Pajarola and E. Gobbetti, "Survey on Semi-Regular Multiresolution Models for Interactive Terrain Rendering," *The Visual Computer*, vol. 23, no. 8, pp. 583–605, June 2007.
- [24] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*. Springer, 1992.
- [25] A. Piva, F. Bartolini, and R. Caldelli, "Self Recovery Authentication of Images in The DWT Domain," *Int. J. Image Graphics*, vol. 5, no. 1, pp. 149–166, 2005.
- [26] P. C. Su, H. J. Wang, and C. C. J. Kuo, "An Integrated Approach to Image Watermarking and JPEG-2000 Compression," *Journal of VLSI Signal Processing Systems, Special Issue on Multimedia Signal Processing*, vol. 27, no. 1-2, pp. 35–53, Jun. 1997.
- [27] W. Sweldens, "The Lifting Scheme: a New Philosophy in Biorthogonal Wavelet Constructions," in *Proc. SPIE, Electronic Imaging, Wavelet Applications in Signal and Image Processing*, vol. 2569. San Jose, CA, USA: SPIE, IS&T, September 1995, pp. 68–79.
- [28] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals; Standards and Practice*. Springer, 2002.
- [29] H. J. Wang and C. C. Kuo., "An Integrated Approach to Embedded Image Coding and Watermarking," in *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing (IEEE ICASSP 98)*, Seattle, WA, USA, May 1998, pp. 3271–3275.
- [30] X. G. Xia, C. G. Boncelet, and G. R. Arce, "A Multiresolution Watermark for Digital Images," in *Proc. IEEE International Conference on Image Processing (IEEE ICIP 97)*, Santa Barbara, CA, USA, Oct. 1997, pp. 548–551.