# Visualisation 3D adaptée par insertion synchronisée de données cachées

Khizar Hayat

**UNIVERSITÉ DE MONTPELLIER II (UM2)**

**ÉCOLE DOCTORALE I2S**
**INFORMATION, STRUCTURE ET SYSTEME**

# P h D T H E S I S

as a partial fulfillment of the requirements of the

**PhD Degree in Science**

by the University of Montpellier II

**Speciality: Computer Science**

Presented and Publicly Defended by

Khizar Hayat

# Scalable 3D Visualization via Synchronous Data Hiding

*Visualisation 3D adaptée par insertion synchronisée de données cachées*

Thesis Supervisor: William Puech
Co-supervisor: Gilles Gesquiere

Realised as a member of the Research Team ICAR at LIRMM

defended on June 22, 2009

**Jury:**

| | | | |
|---|---|---|---|
| *Reviewers:* | Jean-Marc Chassery | - | GIPSA, Grenoble |
| | Luce Morin | - | IETR, Rennes |
| *Supervisor:* | William Puech | - | LIRMM, Montpellier |
| *Co-supervisor:* | Gilles Gesquière | - | LSIS, Marseille |
| *President:* | Jean Claude Bajard | - | LIRMM, Montpellier |
| *Examiner:* | Marc Pierrot Deseilligny | - | MATIS, IGN, Paris |

## Acknowledgment

This work owes its existence to my supervisor Dr. William Puech whose fullest cooperation made it presentable, by the Grace of Allah. His guidance, encouragement and support were available any time I needed. Cool demeanor, humility and patience were his notable qualities during the supervision of this research.

Not to forget my co-supervisor, Dr. Gilles Gesquière, who showed a lot of commitment and attended almost every meeting related to this work and gave always useful suggestions. Gilles, though not working at LIRMM, was a regular visitor for this work.

I would like to express my heartiest gratitude to the honorable reviewers, Dr. Jean-Marc Chassery and Dr. Luce Morin for accepting the task to evaluate this thesis. I thank them for their honest remarks and suggestions to improve the presentation of this work. Gratefully mentioning the names of Dr. Marc Pierrot Deseilligny and Dr Jean Claude Bajard for accepting to be members of the jury to evalute this thesis.

As a member of the ICAR (Image and Interaction) team, I got a lot of encouragement from other members of the group, particularly Dr. Gerard Subsol and Dr. Marc Chaumont, who were in general very kind and the fortnightly meetings were very informative and refreshing. Special to mention is the annual module of ICAR, named *Module Image*, where the monthly lectures provided a platform to know about the works of other researchers in field, from France. We are particularly thankful to one of the visitors, Dr. Phillipe Carré of XLIM, Poitiers, who gave us a useful feedback for this thesis as far as the concept of wavelets are concerned.

I feel indebted to the Higher Education Commission (HEC) of Pakistan as well as the Higher Education Department, NWFP, Pakistan for their support. The role of my laboratory, LIRMM, must not be undermined in this regard. Finally we are grateful to IGN France for the data they provided for this work.

In the end I would love to mention my parents, wife and the rest of my family for showing cooperation, support and patience during the course of the work in perspective. Not to forget, of course, my friends and colleagues, here and back home.

# Contents

# List of Figures

Chapter 1

# Introduction

*La recherche de ces trente dernières années a révolutionné le domaine du traitement des images. Une thématique considérée comme secondaire pendant longtemps, couvre maintenant de larges domaines, tels que la vision par ordinateur, la modélisation et visualisation 3D, la télésurveillance, l'imagerie médicale, la compression et la transmission d'images, la télédétection et d'autres encore. Le traitement des images n'est pas le seul domaine numérique qui puisse témoigner de l'évolution numérique. Il est particulièrement intéressant de constater durant la première décennie du XXIe siècle de l'évolution technologique mondiale, des débits des réseaux, des capacités mémoires et des performances informatique en règle générale. De ce fait, la diversité des supports de travail numérique des clients récepteurs est aussi en expansion. D'un coté, il existe de puissants postes de travail, et à l'autre extrême des appareils portables, comme les* smartphones *avec entre les deux des dizaines d'autres plates-formes. Ces révolutions technologiques doivent donc toujours faire face à ces diversités qui les accompagnent. Ceci reste en permanence un sérieux défi. Ce défi est encore plus flagrant dans le cas d'applications où les données impliquées sont énormes. L'une de ces applications concerne le domaine de la visualisation 3D, où le problème est exacerbé par la participation de données hétérogènes. Une visualisation 3D typique, orientée sur la représentation d'une surface, est basée sur au moins deux types de données : une image 2D d'intensité, appelée texture et une forme 3D rendue sous la forme d'une image de profondeur, ou un modèle 3D ombré, ou un maillage de points. La gamme de données, parfois également appelée carte de profondeur, est une image dans laquelle la valeur des pixels reflète la distance du capteur à la surface par imagerie. Quel que soit le domaine, le processus de visualisation n'est que la superposition de la texture sur la forme 3D.*

*Pour ce type d'applications, le problème à résoudre est donc double du fait de la diversité des clients et de la nature disparate des données. Le souhait de "comprimer une fois et décomprimer de différentes manières", paradigme du codeur JPEG2000 peut fournir un mécanisme afin de faire face à divers types de clients. Le caractère multi-résolution de la transformation en ondelettes nous donne l'évolutivité dont nous avons besoin. Ainsi, dans le cas de visualisation 3D, pour la texture et la carte de profondeur, nous appliquons une transformation en ondelettes discrète (TOD) afin de proposer une solution concernant l'hétérogénéité des clients. En ce qui concerne le deuxième aspect du problème, l'idée principale dans nos travaux consiste à tenter d'unifier les données. Mais cette unification doit conserver un certain degré de synchronisation face au codeur JPEG2000. Le domaine d'insertion des données cachées (IDC) nous offre de nombreux moyens afin de combiner la texture et la*

*carte de profondeur dans le domaine des ondelettes. Le codage pourrait permettre d'obtenir au final un seul et unique fichier standard JPEG2000. Notre travail de recherche est donc à la frontière de différentes disciplines en traitement des images, à savoir, la compression, la modélisation et visualisation 3D, l'IDC et la transmission.*

*Dans une première partie de nos travaux, nous avons opté pour la synchronisation complète de la texture et de la carte de profondeur. Les deux types de données sont soumis au même niveau d'une TOD pour l'insertion. En outre, du fait d'une exigence importante d'imperceptibilité, nous avons, pour l'insertion aveugle, utilisé une méthode basée sur l'utilisation des bits de poids faibles afin de dissimuler la carte de profondeur transformée dans la composante (luminance/chrominance) transformée de la texture dans le processus de codage JPEG2000. Pour le codeur JPEG2000 sans perte, une interruption immédiatement après l'étape de TOD est proposée, alors que pour le cas avec pertes, il est préférable d'attendre l'étape de codage entropique afin de mener à bien l'insertion.*

*Dans une seconde partie, nous avons adapté la synchronisation afin d'améliorer la qualité de la carte de profondeur, qui semble être plus sensible visuellement aux dégradations. Ceci n'a été possible que dans le cas d'une différence importante de tailles entre la carte de profondeur et la texture. Cette approche nécessite donc d'effectuer une intégration de la carte de profondeur dans un sous-ensemble de sous-bandes calculé à partir d'une analyse psycho-visuelle afin d'identifier l'erreur tolérable. Pour augmenter la robustesse, un changement de stratégie d'insertion aveugle est également recommandé en utilisant des approches par étalement de spectres. La proposition de stratégie d'étalement de spectres synchrone adaptative est non seulement robuste, mais en même temps imperceptible du fait que l'insertion est effaçable. Pour finaliser cette partie, nous avons analysé la robustesse aux attaques de nos méthodes, en particulier la robustesse à l'ajout de bruit gaussien et à la compression JPEG.*

*Une fois ce mécanisme de synchronisation mis en place, un problème supplémentaire apparaît quand la visualisation d'une zone intervient entre plusieurs dalles transmises à différentes qualités. Pour illustrer ce problème, nous avons pris en exemple un cas d'utilisation de neuf dalles, avec la dalle centrale de qualité maximale et les dalles périphériques de qualités moyenne ou faible. Une stratégie de lissage dans le domaine de la TOD est proposée pour l'assemblage sans jonction visible entre des dalles de qualités différentes.*

*Cette thèse est organisé en deux parties principales. La Partie I illustre des concepts théoriques. Nous présentons les transformées en ondelettes et le codeur JPEG2000 (Chapitre 2), la modélisation et la visualisation 3D (Chapitre 3), ainsi que des méthodes d'insertion de données cachés (Chapitre 4). Nos contributions sont présentées dans la Partie II. Dans le Chapitre 5, nous présentons une intégration synchronisée des données 3D dans la carte de texture. Dans le Chapitre 6, cette synchronisation est adaptée afin d'améliorer nos approches en terme de qualité visuelle et de robustesse. L'assemblage sans soudure apparente de dalles hétérogènes est présenté dans le Chapitre 7.*

The research of the past thirty years has revolutionized the field of image processing. A field considered to be secondary for a long time, now covers such vast areas like computer vision, 3D modeling and visualization, tele-surveillance, medical image processing, image compression and transmission, remote sensing and many more. Image processing is not alone in testifying this digital evolution. Of particular interest is the revolution being witnessed by the first decade of twenty-first century in the form of network bandwidths, memory capacities and computing efficiencies, in general. Simultaneously, the platform and client diversity base is also expanding. There are powerful workstations on one extreme and hand-held portable devices, like smartphones, on the other; in between are scores of other platforms. The technological revolution notwithstanding, dealing the accompanying diversity is always a serious challenge. This challenge is more glaring in the case of applications where huge data is involved. One such application is the area of 3D visualization where the problem is exacerbated by the involvement of heterogeneous data. A typical surface oriented 3D visualization is based on at least two sets of data: a 2D intensity image, called texture and a corresponding 3D shape rendered in the form of a range image, a shaded 3D model or 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. Irrespective of the field, the process of visualization is nothing but the overlaying of the texture on the 3D shape.

For such applications, the problem is two-fold in terms of the diversity of clients and the disparate nature of the 3D data. The "Compress once, decompress many ways" paradigm of JPEG2000 can provide the mechanism to address diverse clients. The multiresolution character of its discrete wavelet transforms gives us the scalability we need. Thus, in the case of 3D visualization, for the texture and the depth map, we propose to apply a discrete wavelet transformation (DWT) to put forward a solution to the heterogeneity of clients. As far as the second aspect of the problem is concerned, the main idea in our work is to try to unify the data. But this unification must have some degree of synchronization in the face of the structure of the JPEG2000 encoder. The field of data hiding offers us many ways to combine the texture and depth map in the DWT domain. The encoding would thus result a monolithic standard JPEG2000 format file. Our research is, therefore, on the interface of a variety of disciplines in the image processing, namely, compression, 3D modeling and visualization, data hiding and data transmission.

In the beginning of our work, we opt for the full synchronization of the texture and the range data. Both types of data are subjected to the same level DWT for embedding. In addition, with the requirement of imperceptibility in mind, we go for the blind least significant embedding (LSB) embedding in order to hide DWT domain range map in the transformed luminance/chrominance texture components from the JPEG2000 coding pipeline. For lossless JPEG2000, an interruption immediately after the DWT step is proposed, whereas for the lossy case it is better to wait for entropy coding in order to accomplish the embedding.

In the second part, We *adapt* the synchronization, in order to improve the quality of the seemingly more sensitive depth map. This is possible only if there is a

significant difference in size between the depth map and texture. This approach would require the range map to be subjected to lower level DWT than that of texture and embed in a subset of subbands based on a psycho-visual analysis to ascertain tolerable error in the range map. For the robustness, a change of strategy in embedding is also recommended to employ blind spread spectrum embedding. The proposed adaptively synchronous spread spectrum (A4S) strategy is not only robust but simultaneously imperceptible since the embedding is removable. To complement this part, we analyze our methods against robustness attacks which particularly include resistance the addition of Gaussian noise and the JPEG compression.

Once a synchronization mechanism in place, an additional problem appears when the visualization of a zone occurs at the junction of multiple tiles transmitted in varying qualities. To illustrate this problem, we are taking an example use case of nine tiles with center tile at maximum quality and the surrounding tiles have either medium or low quality. A DWT domain smoothing strategy is being proposed for the seamless stitching of these heterogeneous tiles.

This thesis is organized into two main parts. Part I illustrates the theoretical concepts needed for the better understanding of our work. We present mainly the DWT and the JPEG2000 coding standard (Chapter 2), the 3D modeling and visualization (Chapter 3) and information hiding (Chapter 4). Our contributions are being presented in Part II. In Chapter 5, we present the synchronization integration of 3D data in the texture map. In Chapter 6 this synchronization is adapted to improve our approach in terms of visual quality and robustness. Seamless joining of the heterogeneous tiles is being presented in Chapter 7.

# Part I

# STATE OF THE ART

*Dans la première partie de cette thèse, nous présentons les notions fondamentales nécessaires pour comprendre nos travaux de recherche. Dans le Chapitre 2, nous introduisons les ondelettes afin de présenter les transformées en ondelettes discrètes et le standard JPEG2000. Nous détaillons chacune des étapes du codeur JPEG2000 en insistant sur les aspects multi-résolution nous fournissant l'adaptation nécessaire pour les applications visées. Dans le Chapitre 3, la modélisation et la visualisation 3D sont discutées, en faisant référence principalement aux représentations basées surface. Le Chapitre 4, dernier de la Partie I, présente les objectifs des méthodes d'insertion de données cachées. Une attention particulière est accordée à l'insertion des données dans le domaine des ondelettes, puis dans le codeur JPEG2000. Les trois chapitres de cette première partie servent de fondement théorique pour les recherches développées et présentées dans la Partie II de ce document.*

In the first part of this thesis, we are presenting the fundamental concepts needed to understand our research work. In Chapter 2, wavelets are being introduced with the purpose to present the discrete wavelet transform and the JPEG2000 standard. Each step of the JPEG2000 codec is being elaborated with special focus on the multiresolution which would provide the scalability needed for the system in perspective. In Chapter 3, 3D modeling and visualization are discussed with special reference to surface based modeling. The last chapter of Part I, Chapter 4, presents various aspects of information hiding. In this context special attention is being given to wavelet domain embedding that we then extend to JPEG2000. The three chapters included in this part would serve as a theoretical foundation to the practical work being presented in the second part.

Chapter 2

# Discrete Wavelet Transform and JPEG2000

Contents



## 2.1 Introduction

*L'objectif principal de ce chapitre est de présenter de manière générale le concept des ondelettes afin d'introduire le codeur JPEG2000. A partir des séries de Fourier, nous introduisons progressivement les aspects ondelettes en s'appuyant sur des traitements mathématiques. Après l'analyse et l'étude des ondelettes de reconstruction et leur extension à la 2D, l'accent est mis ensuite sur la discrétisation qui nous conduit naturellement à la transformée en ondelettes discrètes (TOD). Cette partie est suivie par la présentation d'algorithmes bien connus de transformées en ondelettes. Finalement, nous présentons l'algorithme de codage JPEG2000 en insistant particulièrement sur la structure, l'adaptabilité et le taux de contrôle.*

The principal objective of this chapter is to present, in general manner, the concepts of wavelets in order to introduce the JPEG2000 codec. Starting from Fourier series, we try to gradually develop the idea of wavelets with some mathematical treatment as a support. After the analysis and reconstruction study of wavelets and their extension to 2D the focus is then shifted to the discretization leading, naturally, to discrete wavelet transform (DWT). This part is followed by presenting some well-known transform algorithms. Finally, we then dissect the state of the art JPEG2000 encoding with an emphasis on the structure, scalability and rate control.

The rest of this chapter is arranged as follows. Section 2.2 develops the concept of wavelets in relevant detail. Some familiar DWT algorithms are explained in Section 2.3 whereas Section 2.4 details the JPEG2000 codec. Section 2.5 concludes the chapter.

## 2.2 Wavelets

Wavelet analysis is basically scale-dependent[1]. This is not a new concept and owes its origin to the early years of nineteenth century when Joseph Fourier introduced his functional analysis and synthesis. According to him any periodic function, $f(x)$ with a period of $2\pi$[2] is the sum of its Fourier series:

$$a_0 + \sum_{k=1}^{\infty}(a_k \cos kx + b_k \sin kx), \tag{2.1}$$

where the coefficients $a_0$, $a_k$ and $b_k$ are given by:

$$a_0 = \frac{1}{2\pi}\int_0^{2\pi} f(x)\ dx, \tag{2.2}$$

$$a_k = \frac{1}{\pi}\int_0^{2\pi} f(x)\cos kx\ dx, \tag{2.3}$$

$$b_k = \frac{1}{\pi}\int_0^{2\pi} f(x)\sin kx\ dx. \tag{2.4}$$

This was a revolutionary leap in the field of mathematics. As far as signal processing is concerned the Fourier series motivated the scientists to introduce what is now known as the Fourier transform (FT) that transforms one complex-valued function of a real variable into another that is the frequency domain representation of the original. The latter depicts the frequencies that are present in the original function (Fig. 2.1) analogous to the notes describing the chord of music. The net result of the FT based decomposition of the function is a set of oscillatory functions.

[1] http://www.amara.com/IEEEwave/IW_overview.html
[2] $f(x + 2\pi) = f(x)$

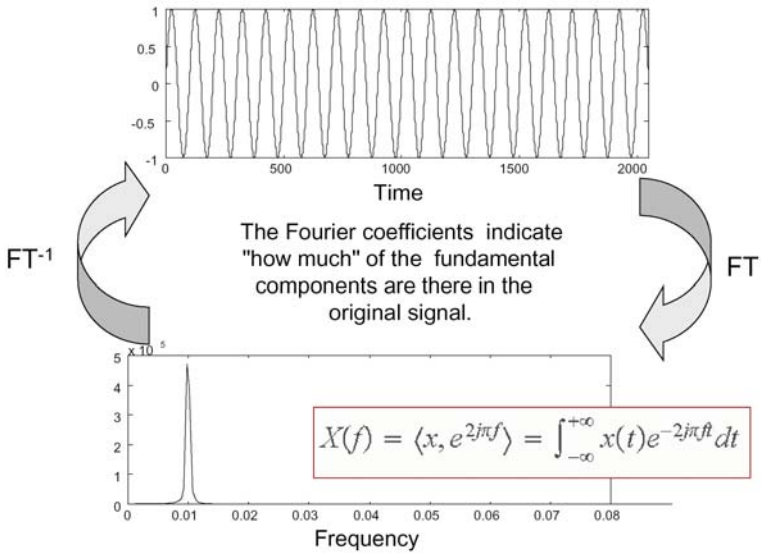
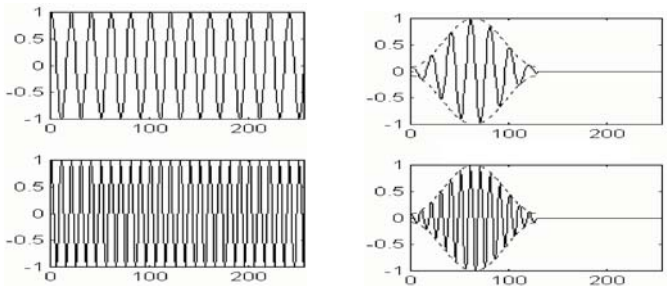



Figure 2.1: Fourier transform in both time and frequency domains.

### 2.2.1 Windowed fourier

With FT, the analysis of the function is infinite due to the absence of temporal information which may not be convenient to approximate sharp spikes. This infinite and non-localized nature of these functions compelled the mathematicians in general to look for a replacement of the sines and cosines bases of Fourier analysis. One solution to this problem is the Fourier Window. In order to limit the temporal support of the elements of the Fourier basis, they are bounded by the application of a window $h(t)$. One thus decompose the function by relying on the basis $h(t-k)e^{2j\pi ft}$, as shown in Fig. 2.2.



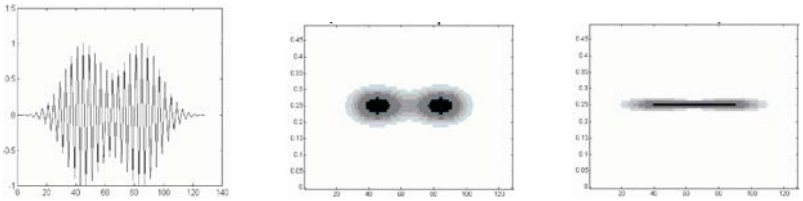
(a) Fourier (b) Windowed Fourier

Figure 2.2: Fourier transform after subjecting to windowing approach.

But the problem with Fourier window is the incertitude in the form of Heisenberg's uncertainty principle:

$$\Delta T \Delta F \geq \frac{1}{4\pi}. \tag{2.5}$$

That is to say a small window will lead to frequency imprecision and a large window may induce a temporal imprecision as depicted in Fig. 2.3. The solution to this



(a) The signal (b) 17 points Window (c) 65 points Window

Figure 2.3: Effect of change in window size on the signal (frequency is drawn as a function of time).

problem could be to search a frequency or temporal domain *slicing* that conforms to the structure of the signal. The idea is to use a variable size and variable position Fourier window and it is here the concept of wavelet transform (WT) comes into play which adapt the window size as a function of the already studied frequencies.

### 2.2.2 Mother wavelet

Wavelet transform[3] decomposes signal into a set of basis functions called wavelets. Wavelets are obtained from a single prototype wavelet $\Psi_{a,b}$, called mother wavelet, by dilations and translations. If $a$ and $b$ are the scaling and translation parameters, respectively then:

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}}\ \Psi\left(\frac{t-b}{a}\right). \tag{2.6}$$

This would change the basis to:

$$W_{a,b} = \langle x, \Psi_{a,b}\rangle = \int_{-\infty}^{+\infty} x(k)\Psi^*_{a,b}(k)\ dk. \tag{2.7}$$

The essence is the multiresolution analysis of the signal components that would calculate successive sub-resolutions. The idea is to imitate the visual mechanism of loss of resolution when the observer's distance from the object increases. This would involve linear decomposition of the signal $s(t)$:

$$\int_{t\in R} |s(t)|^2\, dt < \infty\ or\ \sum_{k\in Z} |s(k)|^2 < \infty. \tag{2.8}$$

The aim is to get a basis of $E$ composed of the elementary functions $\varphi_{n\in Z}$ such that $s \in E$:

$$s = \sum_n \alpha_n \varphi_n, \tag{2.9}$$

[3]Section 2.2.2 is largely based on [Vetterli 1995]

with:

$$\alpha_n = \langle s, \varphi_n \rangle = \begin{cases} \int s(t)\varphi_n^*(t)dt & \text{if } s \text{ and } \varphi_n \text{ are continuous} \\ \sum_k s[k]\varphi_n^*[k] & \text{if } s \text{ and } \varphi_n \text{ are discrete.} \end{cases} \tag{2.10}$$

### 2.2.3 Multiresolution analysis (MRA)

For MRA we construct a family of closed subspaces $Vm$ of $L^2(\Re)$ such that $\cdots \subset V_1 \subset V_0 \subset V_{-1} \cdots$, having the following characteristics:

1. $\cup_{l=-\infty}^{+\infty} V_l$ is dense in $L^2(\Re)$. In other words if $P_{V_l}[s]$ is an orthogonal projection of $s$ on $V_l$ then $\lim_{l\to+\infty} P_{V_l}[s] = s$,
2. $\cap_{l=-\infty}^{+\infty} V_l = \{0\}$,
3. $\forall s$ from $L^2(\Re)$ and $\forall l \in Z$, $s(t) \in V_0 \Leftrightarrow s(2^{-l}t) \in V_l$,
4. $s(t) \in V_0 \rightarrow s(t-n) \in V_0, \forall n \in Z$,
5. $\exists$ a function $\zeta \in V_0$, such that the sequence $\{\zeta(t-\mu)\}_{\mu\in Z}$ is an orthonormal basis of $V_0$.

Hence as the index of scale grows the mesh becomes coarser and the projections of $s$ on $V_l$ contain lesser information. The orthonormal basis $\zeta(t-\mu)_{\mu\in Z}$ of $V_0$ indicates that:

$$\langle \zeta(\cdot),\ \zeta(\cdot,-k)\rangle = \int \zeta(t)\zeta^*(t-k)dt = \delta_{k=0},\ \forall k \in Z. \tag{2.11}$$

We remark that since $\{\zeta(t-k)\}_{k\in Z} \in V_0$, then $\left\{\zeta(2^{-l}t-k)\right\}_{k\in Z} \in V_l$. This leads us to the proposition that $\left\{2^{-l/2}\zeta(2^{-l}t-k)\right\}_{k\in Z} \in V_l$ is an orthonormal basis of $V_l$.

The function $\zeta(\cdot) \in V_0 \subset V_{-1}$ can be expressed as a linear combination of vectors from the basis $V_{-1}$:

$$\zeta(t) = \sum_{k\in Z} h[k]\sqrt{2}\zeta(2t-k) = \left[h * \sqrt{2}\zeta(2.)\right](t) \tag{2.12}$$

and it is this $h[k]$ which gives us the notion of discrete filter:

$$h[k] = \left\langle \zeta(\cdot), \sqrt{2}\zeta(2\cdot - k)\right\rangle = \sqrt{2}\int \zeta(t)\zeta^*(2t-k)dt. \tag{2.13}$$

### 2.2.4 Wavelet analysis

Now the problem arises that how to express the projection of a signal $s \in V_0$ in different spaces of a sampled version $c_0[k] = s(kT_e)$ where:

$$c_0[k] = \langle s, \zeta(\cdot - k)\rangle \rightarrow s(t) = \sum_{k\in Z} c_0[k]\zeta(t-k).$$

A single scale decomposition of the original signal $c_0[k] = \langle s, \zeta(\cdot - k)\rangle$ from $V_0$ thus yields the lowpass $V_1$ component $c_1[k] = \left\langle s, \frac{1}{\sqrt{2}}\zeta(\frac{\cdot}{\sqrt{2}} - k)\right\rangle$ and the highpass $W_1$

component $d_1[k] = \left\langle s, \frac{1}{\sqrt{2}}\psi(\frac{\cdot}{\sqrt{2}} - k)\right\rangle$. The relationship between $c_0[k]$ and $c_1[k]$ can be easily derived:

$$c_1[k] = \sum_{n\in Z} h[n-2k]c_0[n]. \tag{2.14}$$

Taking $x^-[k] = x[-k]$ we have a filter with:

$$(h * c_0)[k] = \sum_{n\in Z} h[k-n]c_0[n], \tag{2.15}$$

we have:

$$c_1[k] = \sum_{n\in Z} h[n-2k]c_0[n] = \sum_{n\in Z} h^-[2k-n]c_0[n] = (h^- * c_0)[2k] = \downarrow_2 [(h^- * c_0)[k]]. \tag{2.16}$$

More simplistically:

$$\begin{cases} c_l = \downarrow_2 (h^- * c_{l-1}) \\ d_l = \downarrow_2 (g^- * c_{l-1}). \end{cases} \tag{2.17}$$

An example sound signal is shown in Fig. 2.4.

### 2.2.5 Wavelet reconstruction

Now to express $c_0$ as a function of $c_1$ and $d_1$, we rely on $V_1 \oplus W_1$ to get:

$$s(t) = \underbrace{\sum_{n\in Z} c_1[n]\frac{1}{\sqrt{2}}\zeta(\frac{t}{2} - n)}_{\text{Approximation at scale 1}} + \underbrace{\sum_{n\in Z} d_1[n]\frac{1}{\sqrt{2}}\psi(\frac{t}{2} - n)}_{\text{Detail at scale 1}}. \tag{2.18}$$

One can derive the relation:

$$c_0[n] = \sum_{n_2\in Z} c_1[n_2]h[n-2n_2] + \sum_{n_2\in Z} d_1[n_2]g[n-2n_2], \tag{2.19}$$

or more generally:

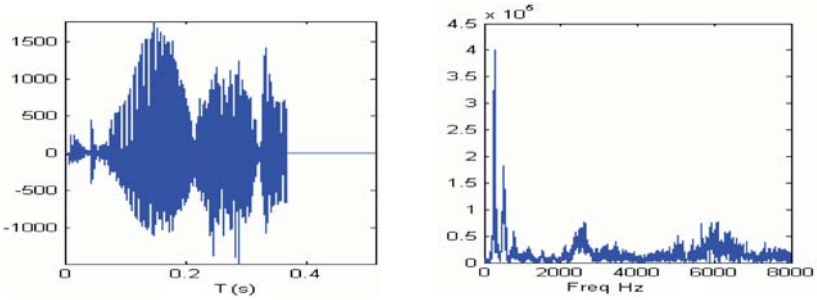
$$c_l[k] = \sum_{n\in Z} h[k-2n]c_{l+1}[n] + \sum_{n\in Z} g[k-2n]d_{l+1}[n]. \tag{2.20}$$

Thus we have the dilation:

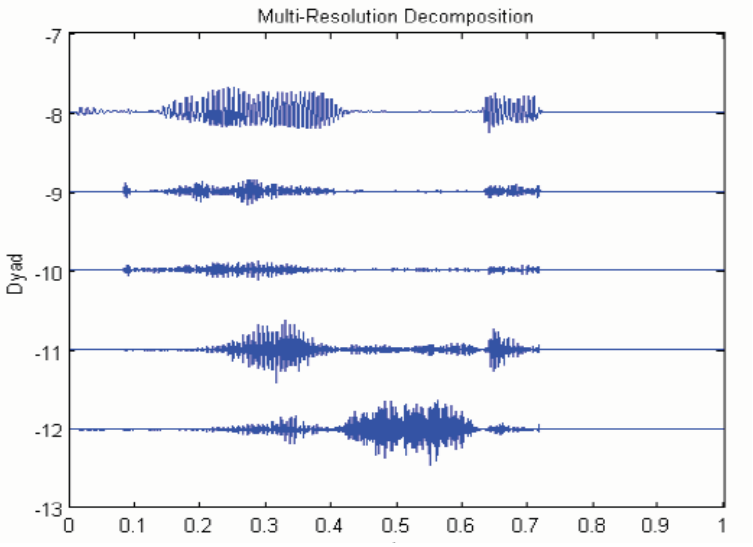
$$c'_l = [\uparrow_2]c_l \Rightarrow \begin{cases} c'_l[2k] = c_l[k] \\ c'_l[2k+1] = 0. \end{cases} \tag{2.21}$$

Hence

$$c_l[k] = [\uparrow_2 c_{l+1} * h][k] + [\uparrow_2 d_{l+1} * g][k]. \tag{2.22}$$




(a) Time Domain (b) Frequency Domain




(c) Wavelet Domain showing various subbands of decomposition

Figure 2.4: An example sound signal subjected to wavelet decomposition.

### 2.2.6 Extension to 2D

The procedure is the same as that for 1D but we have to introduce a dilation matrix $D$:

$$D = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}.$$

With $x(t_1, t_2) \in V_0 \Leftrightarrow x(\frac{t_1}{2}, \frac{t_2}{2}) \in V_1$, with a scaling function $\zeta(t_1, t_2) = \chi(t_1, t_2)$ where $\chi$ is an indicator function of $[0, 1[\times[0, 1[$. The functions $x \in V_0$ are constant (signal 2D sampled) in the zones $[k_1, k_1 + 1[\times[k_2, k_2 + 1[$. The functions $x \in V_1$ are constant over blocks of $2 \times 2$. The complementary space is graphically depicted in Fig. 2.5 wherein the lost information between two successive approximations is obtained by the projection of x into the wavelet space $W$. We have:

$$V_0 = V_1 \oplus \underbrace{W_1^1 \oplus \ldots \oplus W_1^i}_{i=1\ldots(\det D - 1)}. \tag{2.23}$$

The spaces $W_s^i$ are generated by $\left\{\psi_(s,k)^i = \psi(D^{-s}t - k)\right\}_{s=1,k\in Z^2}$. The resultant decomposition is shown in the form of an example in Fig. 2.6.

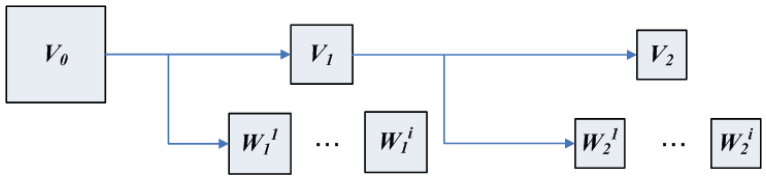



Figure 2.5: Vector Space relation in 2D.

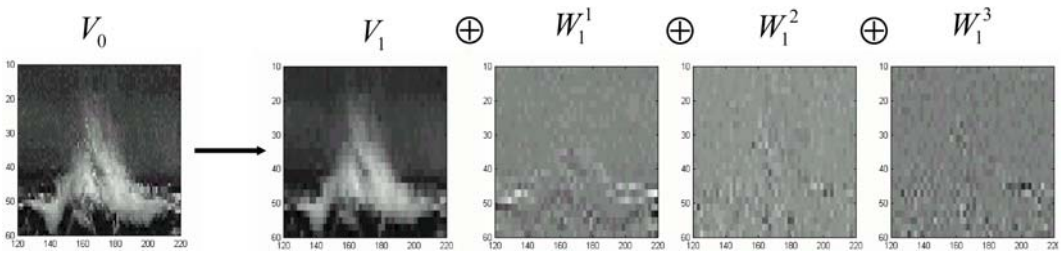



Figure 2.6: Separable dilation matrix for 2D wavelet decomposition.

The function $\zeta(D^{-1}t \in V_1 \subset V_0$ for 2D can be expressed as a linear combination of the vectors of $V_0$:

$$|\det(D)|^{-s/2}\,\zeta(D^{-1}t) = \sum_{k_1\in Z}\sum_{k_2\in Z} h(k_1,k_2)\zeta_0(t_1-k_1,t_2-k_2). \tag{2.24}$$

We have the notion of discrete filter:

$$h'(k_1,k_2) = \left\langle \zeta_0(t_1-k_1,t_2-k_2), \zeta(D^{-1}t)\right\rangle. \tag{2.25}$$

Similarly the function $\Psi^i(D^{-1}t) \in W_1^i \subset V_0$ for 2D can be expressed by the linear combination:

$$|\det(D)|^{-s/2}\,\psi^i(D^{-1}t) = \sum_{k_1\in Z}\sum_{k_2\in Z} g^i(k_1,k_2)\zeta_0(t_1-k_1,t_2-k_2). \tag{2.26}$$

We have the notion of discrete filter:

$$h'(k_1,k_2) = \left\langle \zeta_0(t_1-k_1,t_2-k_2), \zeta(D^{-1}t)\right\rangle. \tag{2.27}$$

In essence we have the following filterbank relationship for decomposition:

$$\begin{cases} c_s = \downarrow_D (h^- \otimes \otimes c_{s-1}) \\ d_s = \downarrow_D (g^{i-} \otimes \otimes c_{s-1}), \end{cases}$$

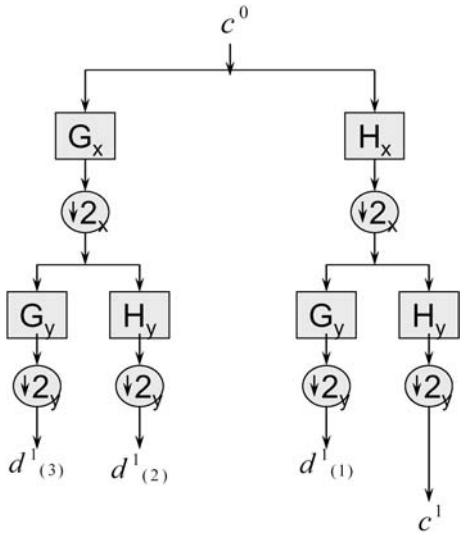



Figure 2.7: A 2D wavelet decomposition schema.

where $\downarrow_D$ refers to the decimation and $\otimes\otimes$ to the filtering. Fig. 2.7 shows a generalized filter structure for the decomposition.

The reconstruction relations are depicted by:

$$c_s = (\uparrow_D c_{s+1} \otimes\otimes h + \left[ \begin{array}{l} (\uparrow_D d^1_{s+1}) \otimes\otimes g^1 + \dots \\ \dots (\uparrow_D d^{|\det D|-1}_{s+1}) \otimes\otimes g^{|\det D|-1} \end{array} \right].$$

## 2.3 The Discrete Wavelet Transform Algorithms

We have seen above that wavelet transform (WT) decomposes a signal into a set of basis functions which are called wavelets. The widespread acceptance of WT in signal and image processing is primarily due to their inherent multi-resolution nature that is well suited for applications with the requirements like scalability and tolerable degradation. All this is possible due to the remarkable self similarity property of various wavelets. Take for example fractal structure of the Daubechies wavelet family [Daubechies 1992] as illustrated in Fig. 2.8

Any transform that samples the wavelets discretely is referred to as a DWT. The DWT splits a component into numerous frequency bands called subbands. For each level, DWT is applied twice, once row-wise and once column-wise and hence four subbands are resulted:

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 ($HH$).

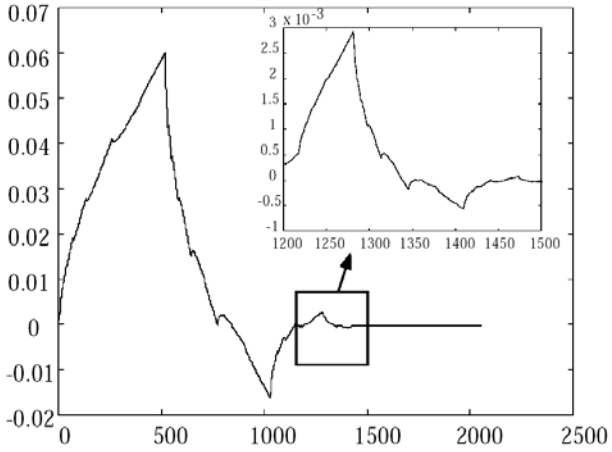



Figure 2.8: The Daubechies mother wavelet, showing the fractal self similarity.


(a) Subband structure (b) Lena (c) Lena at level-2 DWT

Figure 2.9: Discrete Wavelet Transform (DWT).

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 $(0, 1, \ldots R-1)$. 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. 2.9.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. 2.9.b, is illustrated in Fig. 2.9.c.

Wavelet transforms are usually characterized by the symmetry, smoothness, compactness in the form of filter length and orthogonality of the underlying wavelets. Two modes are usually employed for filtering, namely convolution and lifting. The latter mode is the preferred one due to its lesser computational complexity. To improve the efficiency further, the least expensive lifting ap-

proach [Sweldens 1995, Mallat 1998] is employed in a separable rather than non-separable manner. That is to say if the transform is applied row-wise or column-wise in a $1D$ manner. $2D$ 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 $s_0, s_1, s_2, ..., s_{n-1}$ results in the lowpass subband signal $L_0, L_1, L_2, ..., L_{n/2-1}$ and highpass subband signal $H_0, H_1, H_2, ..., H_{n/2-1}$ after the DWT. Some of the most popular of the DWT's are presented in Section 2.3.1, 2.3.2 and 2.3.3.

### 2.3.1 Haar's wavelet transform

Proposed by a Hungarian mathematician Alfred Haar, this transform [Chui 1992] simply pairs up input values while storing the difference and passing the sum. Each recursive repetitions of the pairing process results in the next scale with the ultimate being $n-1$ differences and one final sum. The orthogonal and compact Haar wavelet system has a symmetric scaling but antisymmetric wavelet function with a vanishing moment (a minimum) to characterize it as smooth. Being a limiting case of Daubechies transform, which is being presented next, it is also known as the length-2 Daubechies filter. The lifting approach for calculating 1D Haar's coefficients consists of first averaging the samples in pairs to get $n/2$ averages followed by the difference of each of the average with any one of the coefficient from the original sample pair to have $n/2$ differences. The former is the lowpass and the latter is the highpass result. The lowpass result can again be subjected to the same process for the next level transform. Symbolically, we have:

$$\begin{cases} L_i &= \frac{s_{2i}+s_{2i+1}}{2}, \\ H_i &= s_{2i} - L_i \end{cases} \quad \text{or} \quad H_i = L_i - s_{2i+1}. \tag{2.28}$$

### 2.3.2 Daubechies-5/3 transform

The most popular family of the DWTs is the one by the Belgian mathematician Ingrid Daubechies who postulated it in 1988 [Daubechies 1992]. These transforms are compact orthogonal, antisymmetric if the filter length is more than 2, smooth with the extent of smoothness depending on the length of the filter and its degree of freedom is $N/2-1$, where $N$ is the filter length. The reversible Daubechies-5/3 transform is an integer based transform and that is why it is usually preferred for lossless compression. This simple but effective transform is supported by the JPEG2000 standard. The 1D filter can be defined as:

$$\begin{cases} H_i &= s_{2i+1} - \left\lfloor \frac{1}{2}\left(s_{2i+2} + s_{2i}\right) \right\rfloor, \\ L_i &= s_{2i} + \left\lfloor \frac{1}{4}\left(H_i + H_{i-1}\right) + \frac{1}{2} \right\rfloor. \end{cases} \tag{2.29}$$

As stated above the 2D transformation can be realized by sequentially applying the 1D version row-wise and the column-wise.

### 2.3.3 Daubechies-9/7 transform

For the lossy version we define a new set of variables $s'_0, s'_1, s'_2, s'_3, ..., s'_{n-1}$ where the odd numbered variables ($s'_{2n+1}$) will hold the first stage lifting outcomes and the even numbered ($s'_{2n}$) will hold those of second stage lifting. Let $a$, $b$, $c$ and $d$ are the first, second third and fourth stage parameters, respectively, then for bi-orthogonal 9/7 wavelet transform:

$$\begin{cases} s'_{2i+1} &= s_{2i+2} + a.(s_{2i} + s_{2i+2}), \\ s'_{2i} &= s_{2i} + b.(s'_{2i-1} + s'_{2i+1}), \\ H_i &= \beta'(s'_{2i+1} + c.(s'_{2i} + s'_{2i+2})), \\ L_i &= \beta(s'_{2i} + d.(H_{i-1} + H_i)), \end{cases} \tag{2.30}$$

where $a \approx -1.586134$, $b \approx -0.052980$, $c \approx 0.882911$ and $d \approx 0.443506$. $\beta \approx 0.812893$ and $\beta' \approx 1/\beta$ are the scaling parameters. The boundaries in both types of transforms are handled by utilizing symmetric extension.

## 2.4 The JPEG2000 standard

This section gives a bird's eye view of the JPEG2000 codec. Unless otherwise stated our discussion on this standard is entirely based on [ISO/IEC 2004], [Taubman 2002] and [Adams 2005] as well as a reference from Microsoft[4]. After depicting some history in Section 2.4.1, the stepwise explanation of the codec is illustrated in Section 2.4.2. We then concentrate on the special aspects of the codec, namely, scalability (Section 2.4.3) and rate control (Section 2.4.4).

### 2.4.1 Background

In 1982, the International Organization for Standardization (ISO) and International Telecommunication Union Standardization Sector (ITUT) decided to launch a collaborative effort in order to find out a suitable compression technique for still images. For avoiding redundancy and duplication in their quest, the two organizations formed a working group in 1986 called the Joint Photographic Experts Group (JPEG). After opting for DCT (Discrete Cosine Transformation) in 1988, the international JPEG standard was finally published in 1992 after many deliberations. Various additions were then made in the following years. Since its launching, JPEG has become the most popular image format due to its high and variable rate of compression and its royalty-free nature. But JPEG format has its weaknesses and deficiencies, the foremost being the unrealistic assumption of the independence of $8 \times 8$ blocks that leads to artifacts in most cases. Moreover, with the independent efforts to develop new image compression algorithms it emerged that wavelet based algorithms like CREW (compression with reversible embedded wavelets) and EZW (embedded zero-tree wavelet) provide better compression performance. These are some of the features not present in JPEG. This compelled the approval of the

[4] http://research.microsoft.com/~jinl/paper_2002/msri_jpeg.htm

JPEG2000 project as a new work item in 1996. Work on the JPEG2000 standard commenced with an initial call for contributions in March 1997 with the following key objectives:

1. To allow efficient lossy and lossless compression within a single unified coding framework,
2. To provide superior image quality, both objectively and subjectively, at low bit rates,
3. To support additional features such as rate and resolution scalability, region of interest coding and a more flexible file format,
4. To avoid excessive computational and memory complexity,
5. To ensure that a minimally-compliant JPEG2000 codec can be implemented free of royalties.

Of the alternatives available to the block based transforms the one that got the approval of JPEG2000 [ISO/IEC 2004] 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.

### 2.4.2 The codec

For most of its part, the JPEG2000 standard is written from the decoder point of view, rather than the encoder, though there are enough specifications to assure a reasonable performance encoder. For better comprehension we will follow [Adams 2005] and describe the JPEG2000 from the encoder perspective. Before defining the codec three things are essential to ascertain. Firstly understanding of the source image model in terms the number of components which may vary from 1 to $2^{14}$ but typically it is either one (grayscale) or three (RGB, YCrCb, HSV etc). Secondly, know-how of the reference grid with none of its dimensions exceeding $2^{32} - 1$, e.g. monitor screen at a given resolution is our reference grid. Thirdly, if the image is very large it is divided into tiles of equal dimensions and each tile is treated independently.

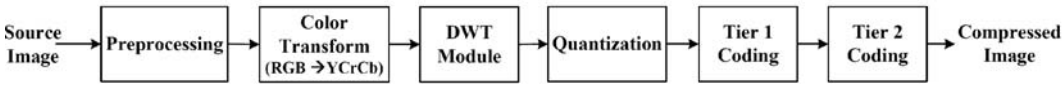



Figure 2.10: A generalized scheme of the JPEG2000 encoder.

A general simplified encoding (Fig. 2.10) can consist of the following steps. It must be borne in mind that not all steps are necessary and some steps can be skipped.

#### 2.4.2.1 Preprocessing

Input data should not be skewed but centered on 0, e.g. for grayscale images the pixel value must be in the range $[-128, 128]$ and thus 128 must be subtracted from all the pixel values in the typical range $[0-255]$. In the decoder this step is canceled by post-processing in the last step. These steps are presented below.

#### 2.4.2.2 Intercomponent transform or color transform

This step involves passing from RGB to YCrCb in what we call the irreversible color transform (ICT) for the lossy case. The RGB to YCrCb transform involves the following equation:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

The inverse would then be:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.34413 & -0.71414 \\ 1 & -1.772 & 0 \end{bmatrix} \times \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix}.$$

For the lossless case an alternative, known as the reversible color transform (RCT) is recommended which is given by:

$$\begin{bmatrix} Y \\ Db \\ Dr \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

#### 2.4.2.3 Intracomponent transform

In image coding, typically, a linear orthogonal or biorthogonal transform is applied to operate on individual components to pass from the space domain to the transform domain. This results in the decorrelation of the pixels and their energy is compacted into a small number of coefficients. This situation is ideal for entropy coding since most of the energy is compacted into a few large transform coefficients: an entropy coder easily locates these coefficients and encodes them. The decorrelation of transform coefficients enables the quantizer and entropy coder to model them as independent random variables. The image community prefers to use block based transforms like DCT and wavelet for obvious reasons. DCT for its quick computation, good energy compaction and coefficient decorrelation was adopted for JPEG. Since the DCT is calculated independently on blocks of pixels, a coding error causes discontinuity between the blocks resulting in annoying blocking artifact. That is why the JPEG2000 coding avoided DCT and opted for the discrete wavelet transform (DWT) that operates on the entire image, (or a tile of a component in the case of large color image). 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, as shown in Fig. 2.9.a. The resultant wavelet coefficients can be easily scaled in resolution as one can discard the wavelet coefficients at the finest M-levels and thus reconstruct an image with 2M times smaller size.

The multi-resolution nature of DWT makes it ideal for scalable image coding. 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) [Daubechies 1998] with the former being lossless and the latter being lossy. Both of these transforms are explained in Section 2.3.

#### 2.4.2.4 Quantization

The coefficients are now quantized to a minimum precision level, necessary for a desired quality. Mathematically, the output signal of quantization step can be described as:

$$s_O = \lfloor |s_I| / \Delta \rfloor \, sign(s_I), \tag{2.31}$$

where $s_I$ and $s_O$ are the input and output subband signals, respectively. The default value of $\Delta$, the quantization step size, is 128. This step is one of the two steps involving information loss; hence $\Delta$ must be set to unity for lossless coding.

#### 2.4.2.5 Tier-1 coding

Each subband is now divided into non-overlapping rectangles of equal size. Three rectangles corresponding to the same space location at the three directional subbands HL, LH, HH of each resolution level comprise a packet. The packet partition provides spatial locality as it contains information needed for decoding image of a certain spatial region at a certain resolution. The packets are further divided into non-overlapping rectangular code-blocks, which are the fundamental entities in the entropy coding operation. A code block must have height and width in the power of 2 and their product, the nominal size - a free parameter -must not exceed 4096. In JPEG2000, the default size of a code-block is 64x64.

After all partitioning the resulting code blocks are independently coded using a bit-plane coder having two peculiarities, viz no inter-band dependency and three coding passes (i.e. significance, refinement and cleanup passes) per bit plane instead of two. The former ensures that each code block is completely contained within a single sub-band, and code blocks are coded independently of one another: improved error resilience can thus be achieved. The latter reduces the amount of data associated with each coding pass, facilitating finer control over rate. Each of these passes scans the samples of a code block in the form of horizontal stripes (each having a nominal height of 4 samples) scanned from top to bottom. Within a stripe, columns are scanned from left to right while within a column, samples are scanned from top to bottom.

The cleanup pass compulsorily involves arithmetic coding but for the other passes it may involve raw coding (involving simple stuffing) too. For error resilience the arithmetic and raw coding processes ensure certain bit patterns to be forbidden in the output. The bit-plane encoding produces a sequence of symbols for each coding pass, some or all of these symbols may be entropy coded through a context-based adaptive binary arithmetic coder. For context selection the state information for either the 4-connected or 8-connected neighbors is taken into account. The context, which is dependent on the bits already coded by the encoder, classifies the bits and signs into different classes expected to follow an independent uniform probability distribution. Let the number of classes be $N$, and let there be $n_i$ bits in class $i$, with the probability of the bits, to take value '1', be $p_i$ then the entropy $H$ of the bit-array according to Shannon's information theory is given by:

$$H = \sum_{i=0}^{N-1} n_i[-p_i log_2 p_i - (1-p_i) log_2(1-p_i)]. \tag{2.32}$$

An entropy coder converts these pairs of bits and the context into a compressed bitstream with length as close to the ideal, i.e. Shannon limit, as possible. There are many such coders and JPEG2000 has borrowed the coder of the JBIG2 standard, i.e. the MQ-coder [ISO/IEC 2000].

#### 2.4.2.6 Tier-2 coding

In this step the coding pass information is packaged into data units called packets by the process of packetization that imposes a particular organization on coding pass data in the output code stream thus facilitating many of the desired codec features including rate scalability and progressive recovery by fidelity or resolution. The packet header indicates which coding passes are included in the packet, while the body contains the actual coding pass data itself. For rate scalability the coded data for each tile is organized into one or more (quality) layers, numbered from 0 to $l-1$, $l$ being the number of layers. The lower layers contain the coding passes having the most important data whereas the upper layers have those with details thus enabling the decoder to reconstruct the image incrementally with quality improving at each increment. Lossy compression involves discarding of some coding passes by not including them in any layer while the lossless case must not discard any coding pass. The code blocks from tier-1 coding are grouped into what are called precincts. For every component-resolution-layer-precinct combination one packet is generated even if it conveys no information at all: empty packets. A precinct partitioning for a particular subband is derived from the partitioning of its parent LL band. Each resolution level has a nominal precinct dimensions which must be a power of two but not exceeding 215. Precinct sizes are to be kept smaller since smaller precincts reduce the amount of data contained in each packet due to the fact that coding pass data from different precincts are coded in separate packets. Thus with lesser data in a packet, a bit error is likely to result in less information loss and higher improved error resilience at the expense of coding efficiency since packet population

is increased. JPEG2000 supports more than one ordering or progressions of packets in the code stream.

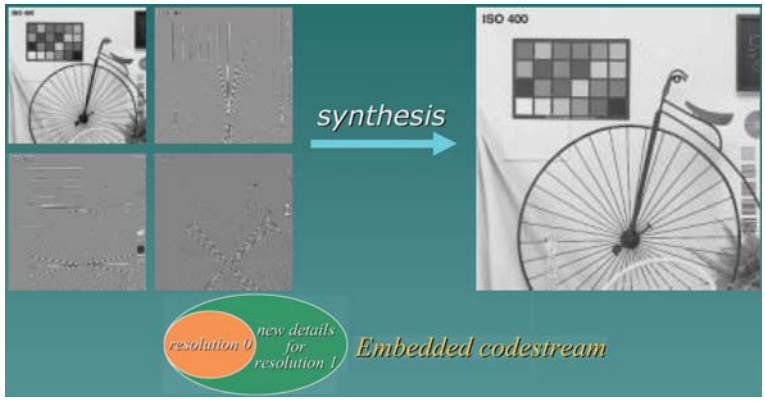
### 2.4.3 The scalability aspect

The JPEG2000 standard offers multidimensional scalability [Taubman 2002] since one can have four different aspects of scalability:

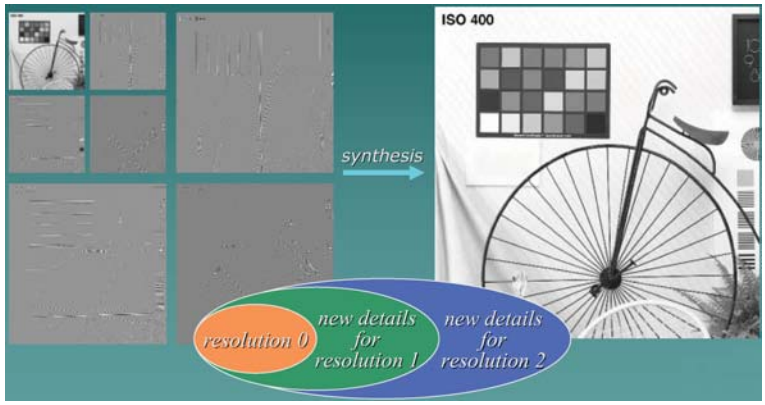
- **Resolution scalability:** The stream progression includes coarse resolution data first, followed by the finer resolutions till the highest resolutions. This scalability is graphically explained in Fig. 2.11.a-b. Resolution scalability gives us control over image size as we can choose it after compression or prior to decompression and even during transmission.
- **Distortion or SNR[5] scalability:** This second dimension of scalability is also called quality or rate scalability since one gets gradually layers of particular SNR in increasing quality as the stream progresses. Fig. 2.11.c illustrates the SNR scalability and one can see that block contributions to each layer can be optimized by the optimal truncation of block bitstream. This allows for custom interpretation of the image quality.
- **Spatial scalability:** With the progression of the codestream one gets a particular region in space followed by the expanded versions, in order.
- **Component scalability:** The components' transfer is also in specific order so that one gets first the grayscale (Y) image followed by other complementary components.

These four dimensions can be combined variously to get still larger number possibilities for scalability. Fig. 2.11.d, depicts the two dimensions of resolution and quality scalability, graphically. More than one scalability dimensions in JPEG2000 necessitates to precise the priorities in the form of information sequencing. The term *progression* is usually employed to identify ordering of bitstream, e.g. a bitstream is quality progressive if the dominant ordering is based on quality layers. Information sequencing is based on the four units of layer (L), resolution (R), Component (C) and precinct or position(P). Based on these we have sequencing like LRCP, RLCP RPCL, PCRL and CPRL. The scalability can give us a number of advantages. For instance the "compress once....decompress in many ways" paradigm gives us the advantage to compress without knowing the needs of end user. In addition one cater a diverse clientele who may have different capabilities in terms of display resolutions, regions of interest, communication rates or other resources. In cases of transmissions with low time out one can even truncate a stream in case of a time out and still get the best possible quality within a deadline. One can even interactively treat a client by changing dynamically the region of interest (ROI) coding, resolution quality and the like.

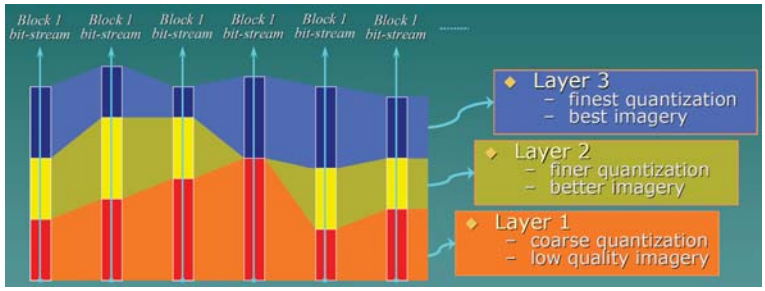
[5]Signal to Noise Ratio



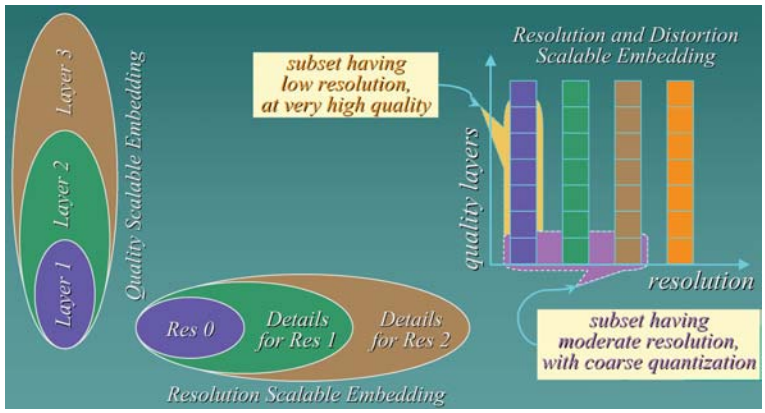

(a) Reconstruction with two successive resolution subbands




(b) Next level added for higher resolution




(c) SNR scalability




(d) Resolution/SNR scalability ensemble

Figure 2.11: Scalability dimensions (Source: author of [Taubman 2002]).

### 2.4.4 Rate control

As already seen there are two sources of information loss in JPEG2000, viz. quantization and discarding of certain coding passes in tier-2 coding. As a consequence there can be two choices for rate control:

1. The choice of quantizer step size
2. The choice of the subset of coding passes that will make up the final bitstream.

With the real-to-real transforms either or both of the above mechanisms are employed but with integer-to-integer transforms only the first one is utilized by fixing $\Delta = 1$.

Apart from the above steps there are other optional operations too. One of these is region-of- interest-coding (ROI) which enables the encoder to code different parts of the same image at different fidelity.

## 2.5 Summary

In this chapter, we have given a brief overview of the evolution of the concept of wavelets and its discrete representation (DWT). Some of the most familiar DWT algorithms were introduced which are specially important in the context of the JPEG2000 codec. Starting from the historical background, the JPEG2000 codec was illustrated stepwise. Interesting aspects of the JPEG2000 coding have been uncovered in the above discussion, the foremost being the scalability and the potential sources of data loss. The multidimensional scalability offers a lot of potential for a heterogeneous client/server environments. This aspect would form the basis of our work on 3D visualization through data hiding. Before that, however, some know-how about 3D visualization and data hiding would not be out of place and that is why we dedicate the next two chapters to their explanation. The data-loss aspect is of particular importance in the data hiding to decide where and when to embed in the cover.

Chapter 3

# 3D Modeling and Visualization

Contents



## 3.1 Introduction

*Les techniques classiques de rendu 2D, à savoir les graphiques vectoriels et matriciels, sont généralement basés sur des plaquages en 3D sur des surfaces ou des volumes. Néanmoins, ces plaquages sont non structurés, dans le sens que, par exemple, les graphiques basés surface adoptent l'approche matricielle pour l'affichage, mais l'approche vectorielle, basée sur des objets, est utilisée pour la représentation, la manipulation et le rendu. D'autre part, en graphique basé volume, la scène est une représentation abstraite d'un volume mémoire, utilisé pour la synthèse, la manipulation et le rendu des scènes 3D. Par rapport à une représentation par volume, la représentation 3D par surface est préférée pour sa vitesse, son faible coût et sa simplicité pour les visualisations où l'intérieur des objets importe peu. Dans ce chapitre, deux applications principales sont mises en avant, à savoir, la visualisation 3D de modèles numériques de terrains et de visages. Nous nous concentrons sur ces deux domaines, en utilisant pour référence des modèles basés sur la surface.*

The conventional 2D rendering techniques, i.e. vector and raster graphics, are generally mapped in 3D to surface graphics and volume graphic, respectively. This mapping is, however, loose in the sense that, for example, surface graphics adopt raster approach for displaying but the object based vector approach for the representation, manipulation and rendering. On the other hand, in volume graphics, scene representation is abstracted by a volume buffer that is employed for the synthesis, manipulation and rendering of the 3D scenes. Hence, the volume graphics are almost entirely raster oriented. As compared to volume representation, 3D surface representation is preferred for its speed, low cost and simplicity for visualizations where the interior is not that important. In this chapter, two principal visualization applications are being put forward, namely terrain visualization and facial visualization. We concentrate on these two areas with reference to surface-based models.

The rest of this chapter is arranged as follows. After explaining the volume and surface-based 3D representation in Section 3.2 we concentrate on the latter in the form of boundary models in Section 3.3. Section 3.4 presents terrain and facial visualizations as representative examples of 3D surface-based rendering. The chapter is summarized in Section 3.5.

## 3.2 3D Spatial Representation

In computer graphics there have been two methods to render 2D objects, namely the vector graphics and the raster graphics. The older of the two, the vector graphics represents the scene by a set of lines or vectors that are repeatedly redrawn to the screen by a random vector generator [Kaufman 1993][1]. Based on vector drawing devices, this technique uses an object-based approach to scene representation, manipulation, and display. It stores a geometric representation of the objects comprising the scene in a display-list which, among other things, has precomputed scene objects to speed up the process of visualization. The objects from the display-list are redrawn each time a screen refresh is needed. A few words about the graphics pipeline will not be out of place over here. Graphics pipeline represents the breakdown of the model from 3D scene to image conversion process into finer steps. This usually involves an operating system based application program interface (API), or library, like OpenGL[2]. A typical pipeline to map 3D scene onto a 2D screen is illustrated in Fig. 3.1[3]. The process let us prepare the 3D scene to be visualized by a device (which can be 2D or 3D). This begins by putting all the objects in the scene in the same coordinate reference system. After that, the viewpoint must be taken into account. This is followed by preparing the view for the device and it includes

[1] http://www.cs.sunysb.edu/∼vislab/projects/volume/Papers/index.html
[2] http://www.cs.cmu.edu/∼jkh/462_s07/02_opengl.pdf
[3] http://medialab.di.unipi.it/web/IUM/Waterloo/node7.html

operations like the projection of 3D on 2D, clipping which consists of cutting those parts of the 3D view which will not be seen in the projected one in the device. Finally, the scene become a raster (an image) which can be drawn on a buffer image (in video card for example), on the device. Almost every step has its own coordinate system (CS) in the form of MCS (model), WCS (world), VCS (viewer), NDCS (normalized device) and device/screen (DSC or SCS). The transformation between two CS is matrix based.

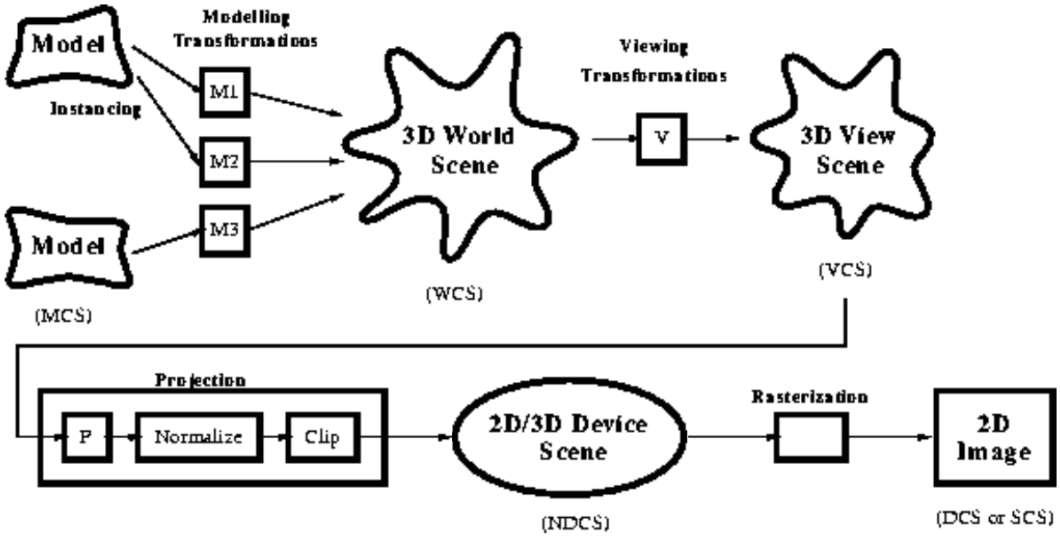



Figure 3.1: A typical graphics pipeline.

The vector approach has the capability to carry out object-based operations on the display-list, besides having the advantage of non-aliasing since the drawings are continuous. The problem, however, is its limited ability to render the interior which makes it suitable for calligraphic drawing only. The raster graphics, on the other hand, represent the scene by a 2D array of pixels stored in a 2D frame-buffer, known as raster. In this approach coloring operation is performed by a point-based renderer that treats the pixels corresponding to the discrete representation of the geometric shape. A video controller is employed to refresh the screen by repeatedly displaying the frame-buffer onto the screen. Raster graphics have ability to present realistic, shaded, and textured surfaces in full color, as well as line drawings. This approach is however memory and computing intensive and not free from the aliasing artifacts, in addition. But the limitations are diluting day by day as the technology progresses and vector graphics are steadily making the way for this approach.

When it comes to 3D graphics, the basic data structures are the same, i.e. the raster and the vector, although they are adapted to 3D. In 3D rasters the volume elements, i.e. voxels, are the building blocks with cell values in the grid accessible by row, column, and level (z). Vectors represent features as a series of discrete linearly connected points. Relevant connections are made between points,

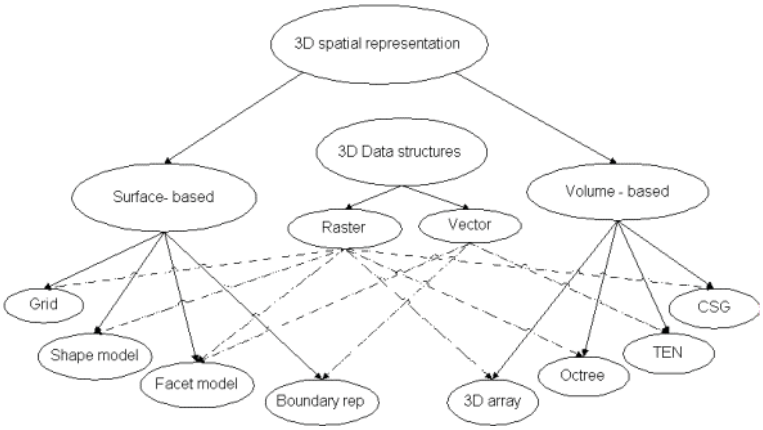
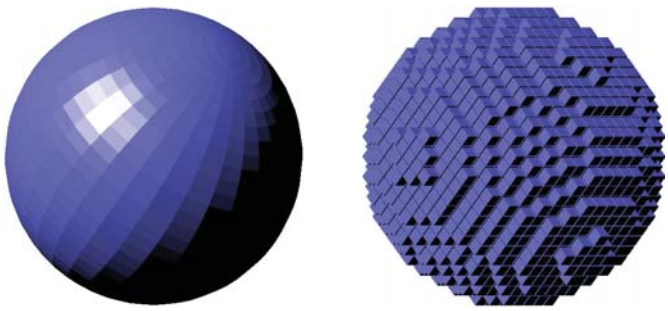



Figure 3.2: 3D spatial object representations

to represent lines or vectors, which in turn join to form polygons. The overall classification of 3D object representation is, however different [Li 1994] as illustrated in Fig. 3.2[4]. The two main approaches for 3D spatial representation are the surface-based and volume based representations[5,6]. As an example one can visually compare the two approaches in Fig. 3.3 showing surface and volume based rendering of a simple sphere.



(a) surface rendered (b) volume rendered

Figure 3.3: rendering of a simple sphere.

### 3.2.1 Surface-based representation

The surface-based techniques adopt raster approach for displaying but the object based vector approach for the representation, manipulation and rendering. Surface rendering visualizes the primitives *indirectly* as their rendering to the screen is preceded by their conversion to an intermediate surface representation. Although raster

[4]http://www.hbp.usm.my/thesis/HeritageGIS/master/research/3D Spatial Representation.htm
[5]http://recherche.ign.fr/labos/cogit/pdf/THESES/RAMOS/These_Ramos_2003.pdf
[6]http://recherche.ign.fr/labos/cogit/pdf/THESES/DELALOSA/These_DeLaLosa_2000.pdf

based in many ways, surface representation shares its advantages and disadvantages with the 2D vector graphics. Hence surface graphics maintain a set of geometric primitives in the form of a display list to represent 3D scenes. The geometric primitives, after transformation and mapping to the screen coordinates, are digitized by a process called rasterization or pixelization. The process involves the application of scan-conversion algorithms and the net result is a discrete set of pixels that is stored in the frame-buffer. This process is repeated by the image generation system every time a change occurs to the scene and viewing or shading parameters and hence the scene description is reprocessed as a whole. That is why it does not take into account the interior of the 3D object, rather it focuses on the the surfaces viewed from a given direction. Surface graphics are thus fast, simple and low-cost. They usually employ OpenGL to specify rendering parameters. With texture mapping one can add the surface realism. On the dark side are characteristics like model hollowness due to negligence of the object's interior, non-support of semi-transparencies and X-ray and limited ability to carry out operations - such as cutting, slicing and dissection. In addition, it is not very easy to model surface-less phenomena like clouds, fog and gas. Surface-based representation can be realized in several ways. Each of these has its advantages, limitations and applicability areas. The most popular ones are as follows [Li 1994, Abdul-Rahman 2008]:

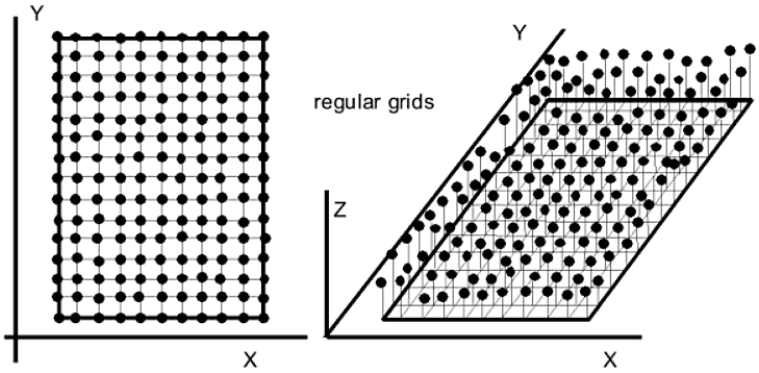
- **Grid:** The basic elements of grids, which may be regular or otherwise, are points with each point having a particular fixed height as shown in Fig. 3.4.a. In this approach the structures are easy to generate while the topology information is implicit. Being particularly suited for geographic information systems (GIS), grid-based methods include techniques like digital mapping, and Digital Terrain Model (DTM). Their main disadvantage is the inability to represent surfaces with variable heights, like a tunnel of a cave, where for the same x and y, there may be several height values.

- **Shape model:** Object representation is based on the surface point derivatives, or slopes, rather than the points as shown in Fig. 3.4.b. The grid point slope can define a normal grid point vector that determines the surface shape. The approach is most suited for 2.5D model reconstruction than for 3D.

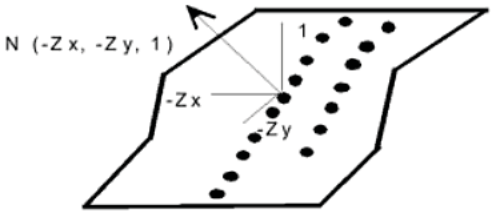
- **Boundary representation (B-rep):** This representation is indirect as it models the bounding surface instead of the entire object. All we have are primitives, like point, edge, face and volume, and the related geometric information. For more details see Section 3.3.

- **Facet model:** A Facet may be referred to as a *surface-by-surface cell* that can have different shapes and sizes; the most famous being the triangle facet. One popular example is the triangular irregular network (TIN), shown in Fig. 3.4.c, that describes a surface by a network of triangle facets of irregular dimensions with each facet having three triangle nodes each represented in the form of x, y and z coordinates. Mostly based on Delaunay triangulations, the triangles may

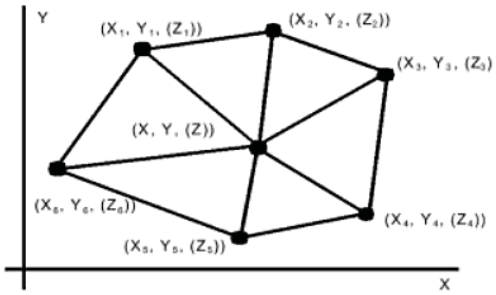
be generated in raster or in a vector domain. Their processing simplicity and structural stability make them suitable for DTM and other terrain surface software. This is more common representation for object modeling. One defines an object with a vertex; vertices are connected with edges. With three vertices, one can obtain a triangle (resp. with four you obtain a quadrangle). The facility with triangular faces is that they can be readily used and rendered with graphic cards.




(a) Grid reprentation with orthogonal and perspective views




(b) surface determination using shape model




(c) 2D TIN model

Figure 3.4: Surface based representations [Abdul-Rahman 2008].

### 3.2.2 Volume-based representation

In volume graphics, scene representation is abstracted by a volume buffer that is employed for the synthesis, manipulation and rendering of the 3D scenes. A volume, also called a 3D image, is a 3D array of point samples termed as volume elements or voxels. These point samples are located at the grid points. The generation of a 2D image from the 3D volume is called volume rendering and it is *direct* process since the visualization does not involve any intermediate representation step. Volume graphics not only shares the benefits of surface graphics but also has additional advantages due its decoupling, uniformity, and atomicity features. Both the exterior and interior of the 3D objects are equally represented. Boolean and block

operations as well as constructive solid modeling are supported in volume graphics. The rendering does not depend on factors like viewpoint, scene complexity or object complexity. The approach is suited for visualization from CT or MRI scan data or scientific simulation data, like Computational Fluid Dynamics (CFD). Volume graphics do have their limitations and the most glaring is its high rendering complexity both in terms of storage and processing. Solutions have been proposed to reduce the complexity, e.g. the multiresolution approach based on 3D wavelets given in [Heurtebise 2006]. Volume-based representation can be realized in a variety of ways, the most popular being the solid modeling techniques like 3D array, octree, Constructive Solid Geometry (CSG) and Tetrahedral network (TEN) which is the 3D version of TIN. In solid modeling, all the primitives are guaranteed to occupy 3D space. Despite having advantages, the problem with solid models is the nonavailability of efficient rendering algorithms and lag in rendering quality behind boundary models. That is why they are transformed to boundary models before rendering, in many cases. We will not expand our discussion on volume graphics as it is beyond the scope of this research work. For further reading we refer [Kaufman 1993].

## 3.3 Boundary Models

In the context of our work we are limiting this discussion to the boundary models wherein the objects are defined by a surface or boundary representation and the models are explicit outside than the inside. Thus it is the bounding surface that is important. The basic components of such models are primitives like vertex, edge, face and even volume combined with the geometric information like edge and face equations or related computation information and vertex coordinates. There are three constraints on a boundary model, namely:

1. the set of faces forms a complete skin of the solid with no missing parts, i.e. shapes must not be open,

2. the faces do not intersect each other except at common vertices or edges,

3. the boundaries of faces do not intersect themselves, i.e. no self-intersection.

Boundary representation may be faceted, elementary, or advanced. The primitives are points, planes and planar polygons represented, implicitly, by their vertices. The elementary version employs planar, quadric, and toroïdal surfaces. The boundaries of the faces are carved by lines, conics, or 4th order curves. Advanced representations employ spline surfaces (B-Spline, Bézier, Non-Uniform Rational B-Splines etc.) as well as those of elementary models with the bounding curves being the spline curves. For more details read [Farin 2002].

Due to the availability of surface information, the boundary models conform closely to the Human Visual System (HVS). They describe the object in terms of geometry and topology. Whilst the geometry depicts the shape and position of the vertices, edges and faces in the form of (x, y, z) coordinates, the topology concerns

the connectivity of the faces, edges and vertices by means of pointers in the data structure. Edges may be lines or arcs while a face is represented algebraically or parametrically. Boundary models can represent a wide class of objects but they have the limitations of being memory intensive and involving complex data structures. Some of the well-known representations are height fields, parametric surfaces and mesh models.

### 3.3.1 Height fields or depth maps

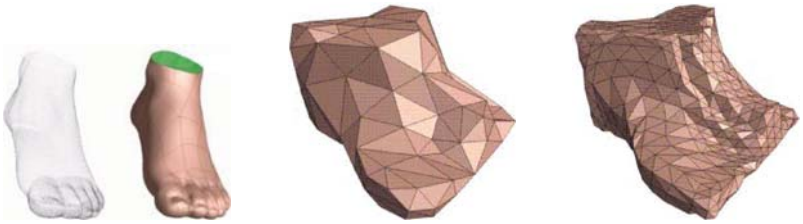
Essentially, it is a set of height samples (3rd dimension) over some planar domain (2D) and thus necessitates an efficient data structure for their representation and display. One of the familiar application is the digital elevation model (DEM) for terrains. They are most naturally represented as discrete functions where $H(x, y) = z$ implies that the point $(x, y, z)$ lies on the surface. The sampling is usually carried out at integer coordinates of a rectangular grid, although the procedure can be extended to scattered points. The problem is, however, their complexity because of the involvement of large number of samples. That is why, these representations are used as raw input of more abstract boundary surface models. In addition, for the same x,y, there is only one z which is a major drawback for representing objects like tunnels in a terrain, for example.

### 3.3.2 Parametric surfaces

Parametric surfaces, like B-splines for example, constitute a mathematical and analytic approach to 3D surface approximation. They can be employed to modeling geometric objects like conic-sections, free-form surfaces, cubes etc. They facilitate complex operations such as rendering and visualization, and thus are gaining popularity in computer graphics and CAD applications. An example of spline meshes is given Fig. 3.5.a. Although these representations are suitable to model complex 3D objects, they are costlier specially for irregular objects due to high number of patches and their interfaces; particularly the problem of continuity management between patches. A simple cure could be the adaptation of these applications to allow, as their input, other boundary representations like polygonal meshes.

### 3.3.3 Polygonal representations

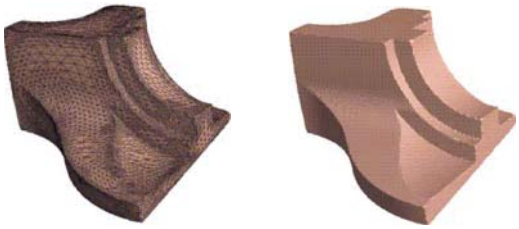
Traditionally, polygon - especially triangle - has been the fundamental primitive for object description as far as graphics hardware are concerned. For obvious reasons, therefore, practical implementations to represent curved surfaces are usually driven by polygonal mesh models. Besides being widely supported, such approximations of sensory range data are simple in execution and sampled surfaces can be approximated with the accuracy needed. Moreover, many methods exist that generate polygonal meshes from existing explicit boundary representations. Often, near planar regions are more frequent in 3D scenes involving large areas, over polygonization is usually avoided by going for large polygons that would considerably reduce the



(a) A mesh of B-spline patches

(b) Very coarse mesh of polygons

(c) Coarse mesh of polygons



(d) Fine mesh of polygon

(e) Very fine polygon mesh

Figure 3.5: Surface rendering: boundary representations.

complexity. In order to conserve resources, in terms of memory, processing and bandwidth, the goal is always not to go beyond the absolutely needed accuracy to model arbitrary scenes. One can have a spectrum of meshes from coarser to fine, as illustrated in Fig. 3.5.b-e. The main problems with the polygonal approximations are the nonphysical significance of points and edges and appearance of numerous small polygons around surface discontinuities. Roudet *et al.* propose a new patch-based multi-resolution analysis of semi-regular mesh surfaces based on surface roughness employing second-generation wavelets [Roudet 2009].

### 3.3.4 Subdivision surfaces

In this type the surface is first defined by a coarse mesh and then by successive iterations newer vertices and edges are added to it until the required smoothness is reached [Bechmann 2007]. The process is carried out according to what is called refinement scheme. Theoretically, the process may require illimited iterations but practically the number of iterations have to be curtailed to avoid complexity that leads to too much density. The subdivision schemes can be seen as the link between discrete models (likes meshes) and continuous ones (based on parametric surfaces) [Sederberg 1998]. Many schemes can be found in this regard but we present some well known[7].

- **The Doo-Sabin scheme** creates a new edge point from the midpoint of each edge and a face point from the centroid of each polygon of the mesh. in the

[7] http://www.holmes3d.net/graphics/subdivision/

end, each vertex in the new mesh is formed as the average of a vertex in the old mesh, a face point for a polygon that touches that old vertex, and the edge points for the two edges that belong to that polygon and touch that old vertex. The new points are then connected. A square in the old mesh, for instance, may create a new, smaller square centered in the old one.

- **The Catmull-Clark scheme** creates a face point for each old polygon by averaging all of the points of the polygon. Accordingly a new edge point is computed from the average of the midpoint of the original edge and the midpoint of the two new face points for the polygons that adjoin the original edge. Finally, new vertices are defined.

- **The Loop scheme** is limited to triangle meshes with every single iteration leads to the splitting of each triangle into four smaller triangles. On each edge, points are constructed which are three eighths of the sum of the two end points of the edge plus one eighth of the sum of the two other points that form the two triangles that share the edge.

According to Balter *et al.*, the wavelets decomposition can be adaptive in order to fit to images and scene contents [Balter 2006]. To ensure time scalability, they propose a representation that is based on a common connectivity for all 3D models, which also allows non-complicated morphing between successive models. With this they ensure visual continuity without any cost escalation.

## 3.4 Example Depth Map Visualizations

The above detailed discussion, in a nutshell, sums up a typical surface-based 3D visualization to rely on at least two sets of data: a 2D intensity image, called texture, with a corresponding 3D shape rendered in the form of a range image, a shaded 3D model and/or 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 [Bowyer 2006]. The underlying terminology may vary from field to field, e.g. in terrain visualization height/depth data is 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 which is overlaid onto the DEM for visualization. Similarly in 3D facial visualization the 2D color face image represents the texture but the corresponding depth map is usually in the form of what is called a 2.5D image. The latter is usually obtained [Conde 2005] by the projection of the 3D polygonal mesh model onto the image plane after its normalization. In a nutshell the basic principal of visualization, be it the face data or terrain data, is the same.

As a whole, 3D object reconstruction is a three step process [Pulli 1997], namely the acquisition, registration and surface reconstruction. The first step concerns range and color data acquisition that involves scanning with the help of multiple

calibrated cameras and some light source. This may also be done by employing specialized camera with which one obtains the depth map. Multiple cameras constitute a stereoscopic system which carry out reconstruction based on multiple images of the same object. In the registration step scans are performed from several viewpoints, in order to obtain complete coverage of the object. In between scans, the object is moved. Since the range data in each view is expressed in the sensor coordinate system the views ought to be aligned, or registered, to estimate a surface from the data and to project the textures to that surface. A solution estimate is thus obtained by interactively matching features and aligning the views in 3D. In this manner a view is registered to another one. The third step is surface model construction, usually through triangle mesh creation.

### 3.4.1 3D terrain visualization

For years, the geospatial professionals thought of map-making as inventing ways to flatten Earth's three-dimensional (3D) surface onto a 2D map[8]. The various methods and projection systems which were developed thus had the limitation of being far from reality albeit accurate. 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[9] and other data sources. 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, multiplayer applications.

#### 3.4.1.1 3D terrain representation

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 a lot lesser heavy files than images. Moreover such a data can be resized without any loss of clarity. Elevations are usually in the form of arrays or grid of terrain heights recorded at regularly spaced horizontal intervals: the result being a DEM. DEM's are used to create 3D images of a landscape when overlaid by the corresponding 2D aerial photos or satellite images thus producing, for example,

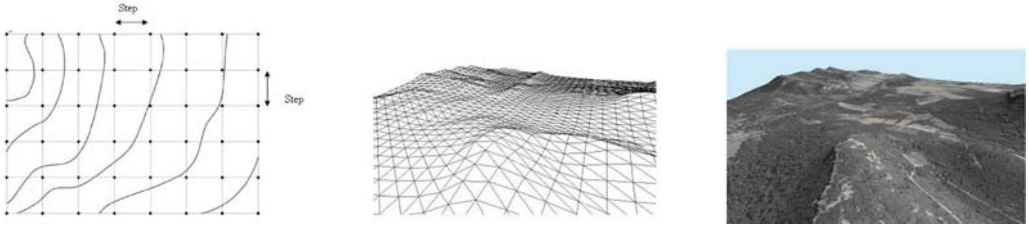
[8] http://www.geoplace.com
[9] http://www.ign.fr

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:

- Height field data which corresponds to the elevation of terrain (Fig. 3.6.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. 3.6.b).
- Different aerial photography can be computed in different spectrum such as visible or infrared. Theses are used to map 3D triangles previously computed (Fig. 3.6.c).

Besides, each set of data has its own coordinate reference system and localization, as the correspondence between these data must be known.




(a) uniform grid corresponding to terrain elevations

(b) 3D triangulated surface linking the elevations

(c) Texture mapping

Figure 3.6: The visualization process.

(a) Vectorial Representation of Carto

(b) Image Extracted from BD Ortho

(c) Image Extracted from BD Scan

Figure 3.7: The IGN data.

In our work we have relied on the data provided by IGN[10] France which consists of vectorial data in the form carto® and raster images in the form BD Ortho® or BD

[10]http://www.ign.fr

Scan® as given in Fig. 3.7. The DEM is usually provided at resolutions expressed in length units, e.g 50 $m$ resolution would mean that two points of altitude are separated by 50 m. 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.

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 of a large area, particularly, if the terrain is composed of hills and valleys. 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 [Fowler 1979] or a progressive mesh representation [Hoppe 1996]. Hierarchical representations of these triangulations have been proposed, thus making it possible to introduce the concept of level of detail [Floriani 1995]. 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.

Another approach consists in breaking up the terrain in a set of nested regular grid of different levels of detail [Bishop 1998, Wagner 2004, Losasso 2004]. This approach has the advantage of allowing an optimal use of the capacities of the graphics cards and thus an increased speed of visualization. The main problem remains to connect the set of grids among themselves without cracks. Several client-server architectures have been proposed to store important number of data often used on several terminals [Baumann 1997]. A strategy consists of cutting a set of images in a nested regular grid [Yu 1996], itself decomposed at various levels of detail [Hansen 1999].

#### 3.4.1.2 The geographical data

A few words about the acquisition and processing of the geographical data would not be out of place here. Since we would be relying on the data provided by the IGN France, this section gives the process[11] outlined by the IGN France International (IGN FI). The raw geographical data is acquired by the following techniques:

- aerial photography,
- satellite imagery,
- scanning, especially for the digitization of legacy maps,
- geodesy which is the study of the earth's shape and dimensions as well as the localization of specific points on its surface,

[11] http://www.ignfi.fr/england/expertiseddonnees01.htm#1

- leveling field survey
- topographic surveys,
- metrology.

The data is then subjected to various forms of treatments through the following:

- Orthophotographs and orthoimages are both digital mosaics, respectively obtained either from aerial photographs or satellite images. These have been corrected geometrically and matched radiometrically.
- Digital terrain models (DTM) and digital elevation models (DEM) are altimetric files are made up of a regular grid of altitudes. The DTM models ground profiles while DEM takes above-ground features (such as buildings and vegetation) into account. These models simulates the 3D environment in a unique manner. They can be calculated from:
  - vectorized contour lines,
  - stereoscopic pairs of aerial or satellite images, by plotting or automatic correlation,
  - radar images pairs, lidar.

  Many types of DEM are available to the public at various resolutions and scales. For example A DEM with 10m resolution, would mean one point for every 10 meters, i.e. the altitudes are spaced at 10 meters. Many methods are available to obtain such a data.
- Spot Image is a provider of geographic information products and services derived from the Spot earth observation satellites. The equipment also includes the vegetation instrument flown on SPOT 4 and 5. The latter satellite has been launched around 2002. The SPOT 1 provided 10 $m$ resolutions which came down to 2.5 $m$ in the case of SPOT 5. In fact, SPOT 5 encompasses topographic mapping, updating, urban planning at scales from 1:25000[12] to 1:10000 as well as defense, environmental and agricultural applications. It consists of the HRG (High Resolution Geometric) with features like achromatic mode (2.5 $m$ and 5 $m$), color mode (10 $m$), strip width of 60 $km$ and oblique viewing capability for a quick coverage. There is the HRS (High Resolution Stereoscopic) aspect having 10 $m$ resolution in achromatic mode, strip width of 120 km, DTM accuracy of 5 to 10 $m$ in relative and 10 $m$ to 15 $m$ in absolute due to the geometric quality of HRS and to the space triangulation processing of IGN. The near future promises the launch of Pleiade which will provide 0.7 $m$ followed, hopefully by Lidar.
- Others like photogrammetric surveys and maps/spacemaps.

[12] 1 cm in the map correspond to 250 meters

#### 3.4.1.3 Geo-data 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 [Gerlek 2004] and GMLJP2 [Lake 2005] for the integration of geographical data with the related aerial photographs. The GeoJP2[13] is a GeoTIFF-inspired method for adding geospatial metadata to a JPEG2000 file. The GeoTIFF specification[14] 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 to describe 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 geo-referencing 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. DEM's 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.

If one talks of specifics, Royan *et al.* have utilized the wavelet based hierarchical LOD models to implement view-dependent progressive streaming for both client-server and P2P networking scenarios [Royan 2007]. An earlier work [Gioia 2004] reconstructs wavelet encoded large meshes in realtime in a view-dependent manner, for visualization, by combining local update algorithms, cache management and client/server dialog.

[13] originally belonged to Mapping Science Inc. and thus had many copyright restrictions, which were later, to a great extent, removed by LizardTech Inc. when it took it over in 2004.

[14] http://www.remotesensing.org/geotiff/spec/contents.html

#### 3.4.1.4 Level of detail (LOD) and DWT

Terrain LOD algorithms can be classified [Losasso 2004], on the basis of the hierarchical structure, into four groups, namely the triangulated irregular networks (TIN), bin-tree hierarchies, bin-tree regions and tiled blocks. The tessellation in the last category is carried out with the square tiles of different resolutions as has been done by Wagner [Wagner 2003]. But the process requires seamless stitching at the boundaries. According to Deb *et al.* [Deb 2006] the view frustum culling technique of [Wagner 2003] is not suited for the terrains with large height variations. They tweak the technique by having the projection on realtime average height of terrain for a client/server terrain streaming to handle heterogeneous clients. The blending factors are calculated on a per tile basis because of the use of a regular tile structure thus reducing the amount of computation. Special techniques based on mipmaps and clipmaps[15] can be found in the literature. Lossasso and Hoppe [Losasso 2004] propose to break the terrains into geometric clipmaps of varying metric sizes which can be used as LODs through a view-centered hierarchy resulting in a simplified spatial and temporal inter-level continuity.

A thresholding scheme based on calculated ground sample distances is proposed in [Tsai 2008] with a nested LOD system for efficient seamless visualization of large datasets. In order to remove the T-junctions they have put forward a mesh refinement algorithm. Ueng and Chuang [Ueng 2000] propose a dike structure between two adjacent blocks, for a smooth blending between two meshes of different LOD. Li *et al.* [Li 2004] combine this approach with heuristics to dynamically stitch the tile meshes together seamlessly. For the texture part they use high resolution aerial photos subjected to the similar LOD mechanism as described for the meshes. The authors of [Larsen 2003] try to avoid popping for low-end users by exploiting the low-level hardware programmability in order to maintain interactive framerates. They claim their work as pioneering as far as a smooth LOD implementation in commodity hardware is concerned.

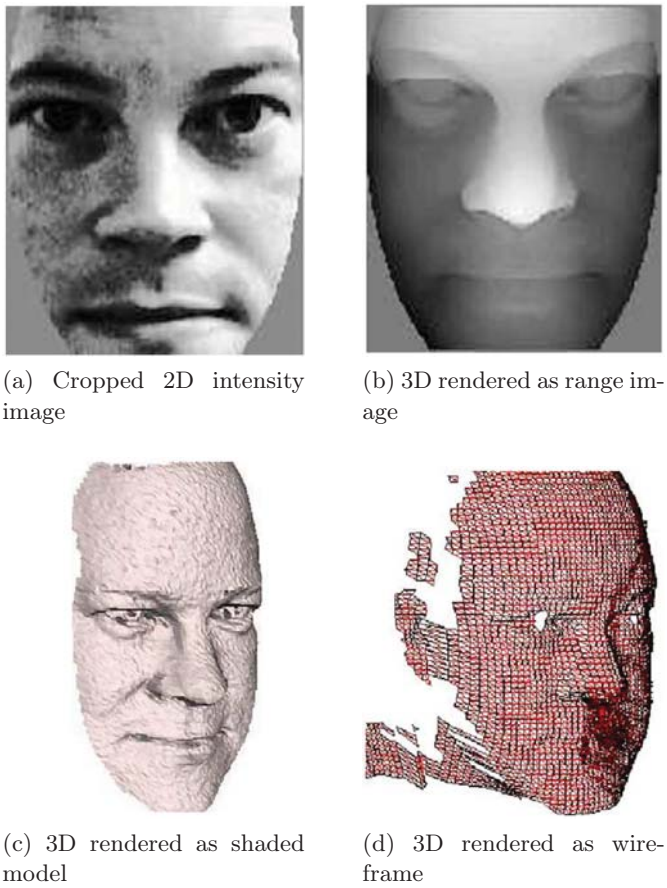
Little attention has been paid to the texture images which require their own LOD structure. There have been proposals, like texture clipmaps [Tanner 1998], but texture tiling is usually preferred. The multiresolution technique, described in [Döllner 2000], presents static LOD terrain models, allowing for combination of multiple largescale textures of different size. The view-dependent texture management technique in [Okamoto 2008] manages multiresolution textures in multiple caching levels - database, main memory and graphics card memory - which the authors claim to be suitable for hardware with limited resources. The approach of [Buchholz 2005] includes dynamic texture loading for memory management and, combine several textures in texture atlases for avoiding too much texture switches [Wloka 2004]. Their computationally complex preprocessing part helps them to significantly reduce runtime rendering overhead. Frueh *et al.*. [Frueh 2004] described

[15] "A mipmap is a collection of correlated images of increasingly reduced resolution arranged, in spirit, as a resolution pyramid whereas a clipmap is an updatable representation of a partial mipmap, in which each level has been clipped to a specified maximum size"[Tanner 1998].

an approach to create a specialized texture atlas for building facades and supports efficient rendering for virtual fly-throughs. The created atlas is static, and different texture resolutions are not considered.

There are specialized approaches like, 4-8 hierarchies of Hwa *et al.* [Hwa 2004] to increase the LODs or Hierarchical Levels of Detail (HLODs) of Lakhia [Lakhia 2004]. A good bibliography can be seen in [Danovaro 2006] and [Buchholz 2005].

### 3.4.2 3D facial visualization



(a) Cropped 2D intensity image

(b) 3D rendered as range image

(c) 3D rendered as shaded model

(d) 3D rendered as wire-frame

Figure 3.8: Example of 2D intensity and 3D shape data [Bowyer 2006].

The 3D shape of the face is often sensed in combination with a 2D intensity image. The 2D image can be thought of as a texture map overlaid on the 3D shape. A range image, a shaded model, and a wire-frame mesh are common alternatives for displaying 3D face data. An example of a 2D intensity image and the corresponding 3D shape is illustrated in Fig. 3.8, with the 3D shape is shown to be rendered in the form of a range image, a shaded 3D model and a mesh of points. In Fig. 3.8, the lighter values are closer to the sensor and the darker values are farther away. The 2D color face image represents the texture but the corresponding depth map is usually in

the form of what is called a 2.5D image. The latter is usually obtained [Conde 2005] by the projection of the 3D polygonal mesh model onto the image plane after its normalization.

Transmitting digital 3D face data in real-time has been a research issue for quite a long time. When it comes to the real-time, two main areas, viz. conferencing and surveillance, suddenly come to mind. In the earlier videoconference applications, the aim was to change the viewpoint of the speaker. This allowed in particular to recreate the simulation replica of a real meeting room by visualizing the *virtual heads* around a table [Weik 1998]. 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 commercial product that offers a true conferencing environment. Some companies, such as Tixeo in France[16], propose a 3D 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 as a gimmick than a useful functionality. This may be true of a videoconference 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. Another application consists in tracking the 3D movement of the face in order to animate a clone, i.e. a model of the user's face [Dugelay 1999]. Another tracking system is based on a multiresolution Active Model search (gradient descent) [Chaumont 2007]. In fact, the transmission of only a small number of parameters of movement or expression can materialize the video through low speed networks. However, recent technology have increased the bandwidth of conventional telephone lines to several Mbps. This has led to a slowing down of research activities on the subject in recent years. Nevertheless, the bitrate limitation still exists in the case of many devices like PDA or mobile phones. It becomes even critical, in particular in remote videosurveillance applications which are gaining increasing economic importance. Some companies offer to send surveillance images on the mobile phones/PDAs of authorized persons but these are only 2D images whereby the identification of persons is very difficult, especially in poor light conditions. Even in scenarios like that, 3D visualization of face would help in surveillance.

## 3.5 Summary

In this chapter we presented an overview of 3D visualization models, although the thrust has been on the surface-based representations. The reason has been the wide popularity, state of the art technological support and lower overheads. Specifically, two surface-based areas of terrain and face visualizations were discussed as examples. Be it facial or terrain visualization, the basic fact remains the same in that more than one set of data are involved with the texture and the range map being the

[16] www.tixeo.com

essential. The need to unify this disparate data by keeping into account the diversity of environments for their use is the basic theme of the research work in perspective. In the context of terrain we did discuss integration efforts like GeoJP2 and GMLJP2 besides dwelling on LOD analysis. For unification we would rely, in our work, on the information hiding and this would be the topic of next chapter. Hence, in the next chapter, the concepts of information hiding are being explored with special reference to the wavelets and JPEG2000.

Chapter 4

# Information Hiding

Contents



## 4.1 Introduction

*Avec l'ère du numérique, certains domaines de la connaissance, qui étaient sur le point de disparaître, sont à nouveau fortement considérés. Parmi ces domaines, nous trouvons la protection des donnés qui peut être abordée, soit par dissimulation, soit par brouillage. La première approche de protection est appelée insertion de donnée cachée et la seconde concerne la cryptographie. L'art de cacher des données existe dans la pratique depuis au moins l'époque des Grecs sous la forme de la stéganographie. D'autres cultures, comme en Chine [Kahn 1967] mentionnent la pratique d'insertion de codes à un idéogramme dans le cadre d'une expédition. L'intérêt récent dans le domaine remonte à l'élaboration par Simmons du célèbre problème des prisonniers [Simmons 1984]. Le scénario se base sur deux détenus d'une prison, Alice et Bob, qui souhaitent mettre en place un plan d'évasion. Leur communication est suivie de près par le gardien de prison, Willie, qui s'il découvre que quelque chose est dangereux alors il isolera complètement Alice et Bob. Pour échapper à la surveillance du directeur de l'établissement, les détenus doivent*

*adopter un système de transmission secret afin de dissimuler leurs messages. Avec l'avènement des médias numériques, de nouvelles voies sont actuellement explorées pour cacher de l'information. Aujourd'hui, si Alice et Bob s'équipent d'un bon système numérique de transmission, Willie risque de ne pas pouvoir arrêter tous les messages cachés. Ces approches de protection, que ce soit par cryptage ou par tatouage, sont basées des algorithmes connus respectant le principe de Kerckhoffs [Kerckhoffs 1883] et une clef partagée entre l'émetteur et le récepteur. Pendant la transmission des données, celles-ci peuvent être attaquées, intentionnellement ou non, ou soumis à des conditions défavorables dans le cas d'une transmission sans fil par exemple. En fonction de l'application, le choix de la technique est déterminé par les conditions de transmission et de réception. Le processus devient de plus en plus complexe à réaliser en fonction des objectifs prioritaires à prendre en compte.*

With the Computer age, some areas of knowledge that were on the verge of extinction, are being again highly regarded. Among these areas, one finds the protection of information whether by its concealment or by its scrambling. The former approach, is called data hiding and the later concerns cryptography. The art of hiding data is in practice at least since the time of the Greeks in the form of steganography. Other cultures, like China [Kahn 1967] refer to the practice of embedding code ideogram at a prearranged place in a dispatch. The recent interest in the field dates back to the elaboration of the famous prisoners problem by Simmons [Simmons 1984]. The scenario is based on two inmates of a prison, Alice and Bob, who wish to establish a plan of escape. Their communication is closely monitored by the prison warden, Willie, who if discovers something malicious would throw Alice and Bob into solitary confinement. To escape the supervision of the warden, both the inmates must adopt a system for secret transmission in order to conceal their messages. With the advent of digital media new avenues has been and being explored for effective information hiding. Today, if Alice and bob are advanced and well-equipped with digital technology, Willie is not behind by any means. These approaches to protection, whether by encryption or by hiding, are based on known algorithms respecting the Kerckhoffs' principle [Kerckhoffs 1883] and a key shared between the transmitter and receiver. During transit, the data can be attacked, intentionally or otherwise, or subjected to adverse conditions in the case of a wireless transmission, for example. Depending on the application, the choice of technique is determined by the conditions of transmission and reception. The process becomes more and more complex depending on the priority objectives to be taken into account.

The rest of the chapter is divided as follows. Section 4.2 lists the fundamental restriction on hiding information followed by a generic classification of the field in Section 4.3. For the sake of the evaluation of quality degradation we are putting forward some well known metrics in Section 4.4. Some embedding techniques are presented in Section 4.5 with special reference to least significant bit (LSB) and

spread spectrum (SS) based embedding. The when and where of data embedding in JPEG2000 is discussed in detail in Section 4.6. This is followed by a context level classification with the accompanied literature survey in Section 4.7 before concluding in Section 4.8.

## 4.2 Constraints

Bender *et al.* [Bender 1996] have underlined the following restrictions and features during the embedding process:

- It is important that the embedding occurs without significant degradation or loss of perceptual quality of the cover.
- For data consistency, original data of the cover rather than header or wrapper must be used for embedding.
- Intelligent attacks or anticipated manipulations such as filtering and re-sampling should not mutilate the embedded data.
- If only a part of the cover is available, the embedded data should still be recoverable, i.e. self-clocking or arbitrarily re-entrant.
- Degradation of the embedded data is always expected as a result of modifications in the cover data. One way to minimize this, may be the employment of error correcting codes.

All of the above points are summed up by Lin and Delp [Lin 1999] when they mention four main factors that characterize the data hiding techniques:

1. **Hiding Capacity:** the size of information that can be hidden relative to the size of the cover.
2. **Perceptual Transparency:** It is important that the embedding occur without significant degradation or loss of perceptual quality of the cover.
3. **Robustness:** the ability of embedded data to remain intact if the embedded image undergoes transformations
4. **Tamper Resistance:** refers to the difficulty for an attacker to alter or forge a message once it has been embedded.

According to the authors, in addition to the above four, other factors like computational complexity of encoding and decoding, resistance to collusion attacks and data security are also important.

## 4.3 Classification

Based on the kind of protection needed there can be three groups of data hiding as stated by the principal author of [Yilmaz 2008] in his Masters thesis:

1. *Steganography* provides protection against detection. Security is the principal issue.

2. *Semifragile watermarking* which protects against forgery is used for authenticating digital material.

3. *Copyright marking* protects against the removal of some hidden mark in a digital content. It must be robust. Based on the origin of the identification mark it can be subdivided to two groups:

   (a) *Fingerprinting* that embeds a different mark for each of the recipients to track transition in the digital content.

   (b) *Watermarking* for embedding the owner/originator's mark onto the digital content. It may be:

      i. *Private*: when only selected people are allowed to check the watermark. It must be robust to malicious attacks.

      ii. *Public*: which makes the images *smart* by embedding a copyright notice into the images. These are not robust to attacks.

Broadly speaking, all of the above comes down to two, namely the steganography and watermarking. For a layman the main difference between these two is that whilst the former must be statistically invisible, no such constraint exists for the latter. In addition, watermarking requires hiding of far lesser data than steganography. Domain-wise, embedding can be carried out in both the frequency 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.

The underlying terminology in both the steganography and watermarking is almost the same. One hears of terms like *cover* or *carrier*, *embedded* and *stego* or *watermarked*, or even simply *marked*. The term cover is used to describe the original, innocent message, data, audio, still, video and so on. The information to be hidden in the cover data is known as the embedded data. The stego data is the data containing both the cover signal and the embedded information. The term warden is after the role of Willie, the jailer, and describes the eavesdropper. Warden may be active or passive depending on the way he monitors. The process of putting the message data to cover signal is known as embedding.

The goal of steganography is to avoid drawing suspicion to the transmission of a hidden message. If suspicion is raised, then this goal is defeated. Discovering

and rendering useless such covert messages is another art form known as steganalysis [Johnson 1998].

## 4.4 Quality Metrics

Before presenting various embedding strategy it would be expedient to present the metrics to compare quality of media before and after embedding. 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 which is given by:

$$PSNR = 10\ log_{10}\frac{255^2}{MSE}, \tag{4.1}$$

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:

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(I\left(i,j\right) - I'\left(i,j\right)\right)^2. \tag{4.2}$$

In order to judge compression performance a measure called compression ratio is utilized:

$$Compression\ Ratio = \frac{Original\ Image\ Size}{Compressed\ Image\ Size}. \tag{4.3}$$

Another measure is, however, what is known as bitrate which is the amount of information stored per pixel.

$$Bitrate = \frac{Compressed\ Image\ Size\ in\ bits}{Original\ Image\ Size\ in\ pixels}\ \text{in bits per pixel (bpp).} \tag{4.4}$$

Strictly speaking, if the embedding is carried out in the transform domain, then the term pixel should be replaced with coefficient inducing 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 coefficient) is recommended.

As far as error measure is concerned, 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: $RMSE = \sqrt{MSE}$.

An interesting measure for assessing the perceptual image quality, to quantify the visibility of errors (differences) between a distorted image ($X$) and a reference image ($Y$), is structural similarity (SSIM) index. It uses a variety of known properties of the HVS and takes into account the similarity with respect to luminance, contrast and structure. The window based SSIM index is given by:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4.5}$$

where $\mu_x$ and $\sigma_x$ are the pixel mean and standard deviation for a window $x$ from $X$ while $\mu_y$ and $\sigma_y$ represent the same for a window $y$ from $Y$. $\sigma_{xy}$ is the covariance

between the two windows and the constants $C_1$ and $C_2$ are designed to avoid unstable values in case $\mu_x^2+\mu_y^2$ or $\sigma_x^2+\sigma_y^2$ or both approach zero. The overall measure MSSIM (mean SSIM) index, over all the windows to which the images are partitioned, is given by:

$$MSSIM(X,Y) = \frac{1}{W}\sum_{j=1}^{W} SSIM(x_j, y_j) \tag{4.6}$$

Where $W$ is the number of windows. For further elaboration we recommend the work by Wang *et al.* [Wang 2004].

## 4.5 Embedding Techniques

Digital data can be embedded in many ways into the images, e.g. sequential, random, non-random (looking for 'noisy' areas of the image, that will attract less attention), redundant, encrypted, non-encrypted etc. Each one of these has its own merits and demerits. Based on the manner to embed, one can classify the image data hiding techniques into many classes based on factors like robustness, imperceptibility [Cox 2008], choice of embedding areas or domain of embedding - spatial or frequency - and the likes. For example, they are classified as follows in [Lenti 2000]:

- Least significant bit (LSB) insertion
- Public key embedding
- Transform domain based embedding
- Masking and filtering techniques

A more comprehensive classification can be found in [Meerwald 2001a][1]. For our convenience, however, we are classifying these into three categories on the basis of popularity and tradition, namely the least significant bit (LSB) insertion, spread spectrum embedding and innovative methods.

### 4.5.1 Least significant bit (LSB) embedding

LSB techniques embed the message bits directly into the least-significant bit plane of the cover image in a deterministic sequence [Lin 1999]. Pixel allocation may be sequential or a pseudo-random number generator (PRNG) [Fridrich 2003] may be employed using some secret key. Probably the most popular, LSB embedding techniques embed data bits in the least significant bits of the image under the assumption that the resultant change would be highly imperceptible due to obvious limitations of human visual system (HVS). A significant amount of information can be embedded without visible loss of quality of the cover image. Fig. 4.1 shows the process of spatial domain LSB embedding when one bit each is embedded in all the LSB's of red green and blue portions of pixel. It must be noted here that with

[1] http://www.cosy.sbg.ac.at/~pmeerw/Watermarking/MasterThesis/wm_thesis.zip

JPEG and JPEG2000, the YCrCb space, rather than the RGB space, is utilized for embedding.

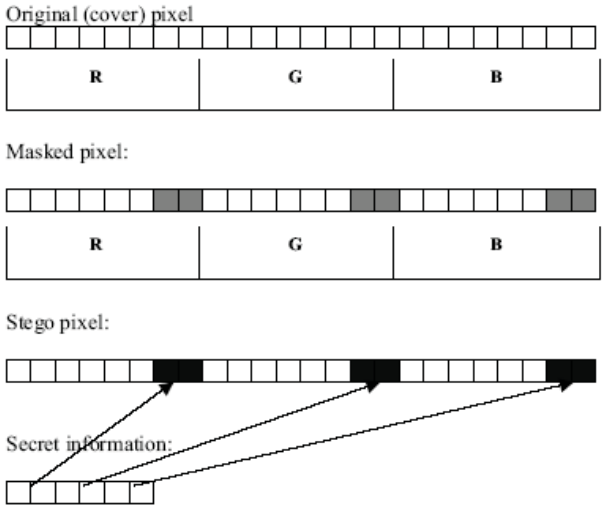



Figure 4.1: An example LSB embedding in the RGB space ( [Lenti 2000])

The LSB based embedding techniques have many advantages over other techniques. Some of these are given below:

1. popularity,
2. easy to comprehend,
3. high perceptual transparency,
4. low degradation in the image quality,
5. more and more commercial software support.

These techniques are not without their limitations. The foremost being:

1. low robustness to malicious attacks,
2. vulnerable to accidental or environmental noise,
3. low tamper resistance.

An example shown in Fig 4.2 illustrates a spatial domain LSB embedding, in a $256 \times 256$ gray scale image, the *girl*, with one bit embedded per 16 pixels (1/16 bpp) using pseudorandom allocation. One can see that there is little difference between the original and the embedded image as appear to the naked eye; a fact confirmed by the PSNR which is 63.21 $dB$.

Note that embedding may not be restricted to LSB and other bit positions may also be used, e.g. the SSB-4 method [Rodrigues 2004] embeds in the fourth LSB of

(a) Original (b) LSB Embedded (c) Difference

Figure 4.2: The Girl image subjected to 1 bpp LSB embedding.

the pixel and then modifies the other bits of the pixel in order to comply with fact that the difference between the new pixel value and the previous one must be equal or smaller than four.

### 4.5.2 Spread spectrum (SS) embedding

This frequency domain technique is a kind of redundant embedding wherein the message is encoded with a sequence of symbols. The transmission of each symbol in the form of a signal, called a chip [Cox 2008], follows a temporal sequence. The chips are, usually, pseudo-random sequences of 1's and 0's. In the frequency domain, they are spread across a wide range of frequencies. Hence, if the signal is subjected to some operation that affects only a fraction of the frequencies, like band-pass filter or band-limited noise addition, the chips may still be identifiable. Fig. 4.3 illustrates

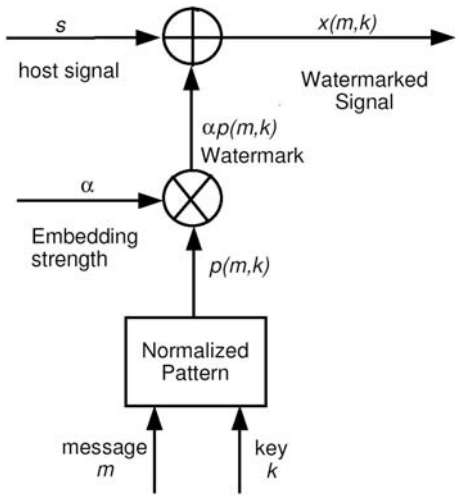



Figure 4.3: A General SS encoding scheme

a generalized SS encoding procedure[2]. The message $m$ encrypted with a key $k$, yields a normalized pattern $p(m,k)$. The embedding strength, which is denoted by $\alpha$, after the mark generation operation yields $\alpha p(m,k)$ that on embedding into the host signal $s$ results in the embedded signal $x(m,k)$. The decoding process may involve correlation analysis, image restoration techniques and/or error control coding [Marvel 1999]. For decoding, usually a test statistics $t(y,m,k)$ is defined and the aim is to find $m$ that maximize $t(y,m,k)$. Some example statistics are:

1. Likelihood ratio test statistics:

$$t(y,m,k) = \frac{p(y,k|m)}{p(y,k|0)},$$

   which may implemented in case the attack channel is known.

2. Correlation statistics:

$$t(y,m,k) = y.p(m,k),$$

   which is well-suited for blind systems and that is why it is the most popular.

3. Correlation statistics:

$$t(y,m,k) = (y-s).p(m,k),$$

   which is suited for nonblind systems.

For better results, $\alpha$ can be made dependent on local properties of $s$.

As an example take again the image from Fig 4.2.a. and subject to SS embedding at various strengths ($\alpha$). A message is being embedded in such a way that one bit requires a $8 \times 8$ pixel block. As a result one gets the embedded images given in Fig. 4.4 where the deterioration in quality ranges from a PSNR of 48.13 $dB$ to 36.10 $dB$. This distortion is far greater than than the one observed for the LSB embedding example in the Section 4.5.2 where a four times bigger message was embedded. The difference images corresponding to Fig. 4.4 are given in Fig. 4.5.

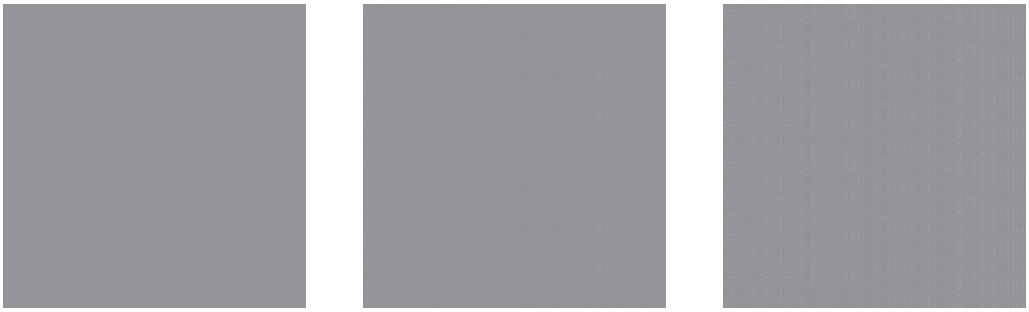
### 4.5.3 Innovative methods

Several techniques have been reported in the literature that are either loosely based on the above two techniques or independent in their approach. Even the blending of cryptography with data hiding has not been a rare practice and many consider this to be quintessential for the security. According to Lenti [Lenti 2000] Public Key Embedding requires the pre-existence of a shared secret key to designate pixels which should be tweaked. Thus, both the sender and the receiver must have this secret. The idea of private/public key pair does not work since the eavesdropper can use the public key to sabotage the whole affair. Transform embedding techniques embed the data by modulating coefficients in a transform domain, such as Discrete

[2]http://www.isit2006.org/Tutorial_TUT-2.asp

(a) $\alpha = 1$ (48.13 $dB$) (b) $\alpha = 2$ (42.11 $dB$) (c) $\alpha = 4$ (36.10 $dB$)

Figure 4.4: SS embedding at various strengths in the Girl image.



(a) $\alpha = 1$ (b) $\alpha = 2$ (c) $\alpha = 4$

Figure 4.5: Difference brought by SS embedding in the *girl* image at various $\alpha$.

Fourier Transform (DFT), Discrete Cosine Transform (DCT) of JPEG or DWT. Modifying the transform coefficients provides more robustness to the compression (especially to lossy), cropping, or some image processing. A JPEG-based DCT domain embedding example is presented in Fig. 4.6 with one bit embedded per $8 \times 8$ DCT block. The observed PSNR was 49.21 $dB$. Again the capacity is small and deterioration is more as compared to LSB-based embedding.

(a) Original (b) DCT Embedded (c) Difference

Figure 4.6: The *girl* image with DCT domain embedding from JPEG encoder.

In sharp contrast to other embedding techniques, the masking or image-adaptive techniques embed information to perceptually significant areas of the image [Cox 2008]. The use of *significant areas* makes these techniques very robust. Masking refers to the phenomenon where a signal can be imperceptible to an observer in the presence of another signal -referred to as the masker [Lin 1999]. The phenomenon of camouflage is manifestation of this human weakness. The image must be analyzed in advance for the information to determine appropriate regions to place the message data so that it is camouflaged in the environment. The MIT's Patchwork algorithm selects random $n$ pairs $(a_i, b_i)$ of pixels, and increases the brightness of the brighter pixel $(a_i)$ by one unit while decreasing the brightness of the other $(b_i)$. The algorithm had shown a high resistance to most nongeometric image modifications [Lenti 2000]. For small messages redundant pattern encoding can be employed to repeatedly scatter hidden information throughout the cover image. For qualitative control over robustness one can also exploit the coefficients having known robustness for embedding. An intelligent approach could be to either invert distortions in the decoder or pre-invert it in the embedder. All these techniques are well-documented in [Cox 2008].

## 4.6 The When and Where of Embedding in JPEG2000

Being an active research area for the last two decades, data hiding is now an established field and that is why a lot has been written about it [Cox 2008]. We, therefore, focus on the literature about wavelet-based data hiding which is again very extensive and one is compelled to be brief and limit oneself to JPEG2000, as far as possible. Looking at the structure of JPEG2000 codec, as explained in Chapter 2 makes one think about when and where to interrupt the coding flow in order to embed the message. Theoretically, you can interrupt the codec anywhere for embedding but, at the periphery, the embedding capacity is lower, accompanied by relatively higher distortion. Fig. 4.7 illustrates the potential interruption stages during the JPEG2000 coding to embed data in the to-be-encoded image[3] [Hayat 2009c]. Every type of intervention has its advantages and limitations.

- Embedding immediately after the DWT step would have the advantage of larger word size of the coefficients leading to high capacity. All the components are easily available one can allocate coefficients at will. This strategy may be especially convenient for JPEG2000 in lossless mode. The problem is however that steganalysis is easier since there is a high probability of unusual coefficient values. This is particularly true of coefficients belonging to high frequency subbands. Moreover embedding must be at least robust enough to resist the ensuing steps of quantization and T1-coding.

- Just after quantization, one can embed in the clipped coefficients with reduced capacity. The overhead of anticipating the loss, due to quantization, is elimi-

[3] http://www.cs.nthu.edu.tw/∼yishin

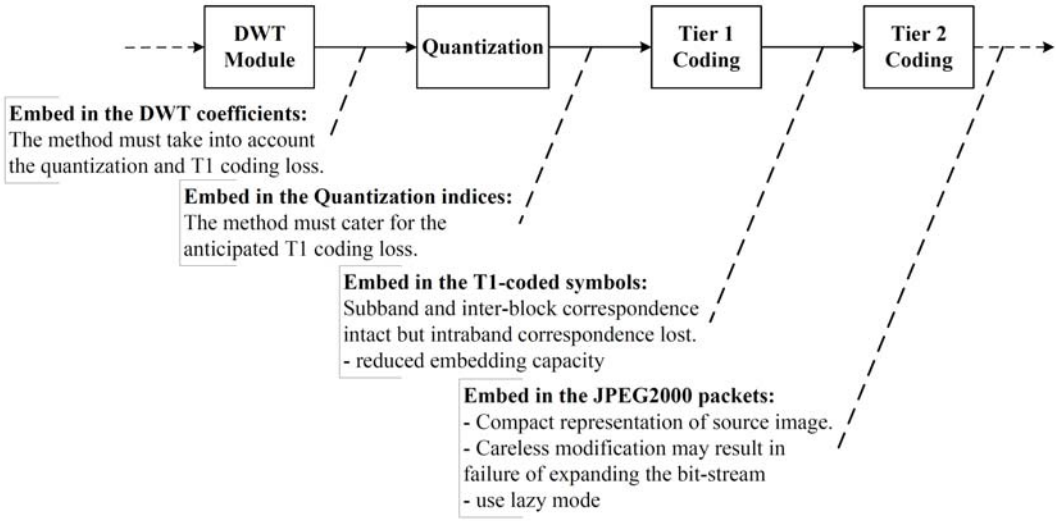



Figure 4.7: Various embedding points in the JPEG2000 coding pipeline.

nated with this type of embedding. Strictly speaking, however, the technique is the same as the last one and shares its pros and cons.

- As already stated, T1-coding operates on the independence of block and comprises bit-plane coding with three passes in each bit-plane, namely significance, refinement and cleanup passes. This followed by the arithmetic coding (MQ coder). One way to intervene is to take advantage of the fact that the partitioned code blocks are coded independently using the bit-plane coder thus generating a sequence of symbols with some or all of these may be entropy coded. The T1 coded symbols from a given block vary in energy and the low index symbols are more energetic than the higher index ones. What can be done, for example, is to use the least energetic of these symbols, from the tail of the stream for each code block, for embedding implying non-random allocation. There is, however one problem in that the T1 coded symbols have smaller word size resulting in smaller embedding capacity and higher rate of distortion in quality as a result of embedding. This policy is not, however, advised in the lossless case since wordsizes of the coefficients are longer at the earlier steps thus leading to lesser distortions as result of embedding. In addition the embedding capacity is limited for such an embedding strategy and the rate of degradation is still larger.

  An alternative approach could be to go for lazy mode and bypass arithmetic coding for most of the significance and refinement passes, except 4 MSBs, however. There would be no substantial benefit from entropy coding in such a scenario. The refinement pass carries subsequent bits after the MSB of each sample hence modification should not cause problems. The significant bits would act as masking which should make the modification of the subsequent

bits less obvious. Hence the lazy mode mostly involves raw coding. Care must be taken in selecting potential raw coded magnitude refinement passes for embedding, otherwise there may be high degradation in quality. This may involve close examination of the bit-planes. The limitations are escalation in the size of the coded image and suspicion in the form of unusual bit stuffing and unusual appearance of error resilience marker.

- Subsequent to lazy mode encoding one can also embed in the T2-coded bitstream. This approach may be simple but has problems in the form of low capacity and high degradation wherein careless modification may result in failure of the expanding bitstream. The easiest way for a novice may be to intervene here and that is why this intervention may be popular but this popularity makes it an easy target of steganalysis.

## 4.7 Classification of JPEG2000-Based Data Hiding Methods

Wavelet based information hiding can be classified in various ways depending on the criteria employed. Many criteria, like decomposition strategy, embedding technique, goal, application, extraction method and many others can be employed for classification. But for our purpose we will use classification where we will be taking into account the *when and where* factor to embed in the JPEG2000 coding pipeline. We call this a *context-based criterion* for classification. Before the advent of JPEG2000 many methods existed in the literature. A very elaborate compilation of these methods can be found in the form of [Meerwald 2001a]. Not all of these methods are compatible with the JPEG2000 scheme. According to [Meerwald 2001c], data hiding methods for JPEG2000 images must process the code blocks independently and that is why methods like inter-subband embedding [Kundur 1999] and those based on hierarchical multi-resolution relationship [Kundur 1998] have not been recommended. In the same breath they reject the correlation-based method [Wang 1998] as well as non-blind methods. The reason for they give 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.

The fact to classify in the context of JPEG2000 is driven by its coding structure as well as the multiresolution character of DWT.

### 4.7.1 Embedding in the DWT coefficients

We further classify these methods into lowest subband methods, high or detail subband methods, trans-subband methods and methods exploiting the coefficient relationships in subband hierarchy.

#### 4.7.1.1 Lowest subband methods

Embedding in lowest subband coefficient is suited for cases where the image has to be authenticated at every resolution level. The problem is however the size of the subband which is a dyadic fraction of the total, thus leading to reduced capacity. Moreover, since most of the energy is concentrated in the lowest subband, the embedding would definitely lead to low perceptual transparency. As an example of this type of embedding can be found in [Xiang 2007] which uses the invariance of the histogram shape to rely on time-frequency localization property of DWT to propose a watermarking scheme that is resistant to geometric deformations. A geometrically invariant watermark is embedded into the low-frequency sub-band of DWT in such a way that the watermark is not only invariant to various geometric transforms, but also robust to common image processing operations.

#### 4.7.1.2 High or detail subband methods

In contrast to low subbands, higher subbands may provide larger capacity. But this is accompanied by escalation in the final image size as the detail subband coefficients hover around zero. 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 [Noore 2007]. Instead they propose to redundantly embed information in all the higher frequency subbands. There are methods for embedding invisible watermarks by adding pseudo-random codes to large coefficients of the high and middle frequency bands of DWT but these methods have the disadvantage of being non-blind [Xia 1997, Kundur 1997]. An additive method transform the host image into three levels of DWT and carry out embedding with the watermark being spatially localized at high-resolution levels [Suhail 2003].

#### 4.7.1.3 Trans-subband methods

To avoid high computational cost for wavelet-based watermarking Woo *et al.* propose a simplified embedding technique that significantly reduces embedding time while preserving the performance of imperceptibility and robustness by exploiting implicit features of discrete wavelet transform (DWT) subbands, i.e. the luminosity information in the low pass band, and the edge information in the high pass bands [Woo 2005]. 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 [Kong 2004]. One blind method 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 [Liu 2006]. 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 concept of Singular Value Decomposition (SVD) has been employed [Yavuz 2007] for their watermarking scheme wherein the $m \times n$ image matrix

$A$ is decomposed into a product of three matrices ($USV^T$); the $m \times m$ matrix $U$ and $n \times n$ matrix $V$ are orthogonal ($U^TU = I$, $V^TV = I$) and the $m \times n$ diagonal matrix $S$ has $r$ (rank of $A$) nonzero elements called singular values (SVs) of the matrix $A$. The SVs of the watermark are embedded into SVs of the LL and HL subbands of the cover image from level-3 DWT domain while components of $U$ matrix of the watermark are embedded into LH and HH subbands. In extraction, first the similarity of extracted $U$ matrix is checked with the original one. If it is found similar, the watermark is constructed by using extracted SVs and original $U$ and $V$ matrices of the watermark. Another DWT-SVD based method employs particle swarm optimizer (PSO) for watermarking [Aslantas 2008]. Agreste *et al.* put forward a strong wavelet-based watermarking algorithm, called WM2.0 [Agreste 2007]. WM2.0 embeds the watermark into high frequency DWT components of a specific sub-image and it is calculated in correlation with the image features and statistical properties. Watermark detection applies a re-synchronization between the original and watermarked image. The correlation between the watermarked DWT coefficients and the watermark signal is calculated according to the Neyman-Pearson statistic criterion just like the blind chaotic method of DWT oriented watermarking [Dawei 2004]. The SS method by Maitya *et al.* embeds watermark information in the coefficients of LL and HH subbands of different decompositions [Maitya 2007]. In two-band system, to increase embedding rate, the cover image is decomposed in different directions using biorthogonal wavelets (BiDWT). For embedding each watermark symbol bit, pseudo-random noise (PN) matrix of size identical to the size of LL subband coefficient matrix is generated and modulated by Hadamard matrix. This modulated code pattern is used to embed data in the LL subband while its bit-wise complement gives an orthogonal code pattern which is used for data embedding in the HH subband. To decode message bit for binary signaling, two correlation values (one from LL and the other from HH) are calculated. The overall mean of these correlation values serves as the threshold for watermark decoding.

#### 4.7.1.4 Methods exploiting subband coefficient hierarchy

Such methods may suitable for embedding resolution scalable messages. An example is image fusion when a small image is embedded in the larger one. Similarly 3D meshes can be embedded by hiding coarse meshes in low and finer details in high frequency coefficients. One can employ data structures like the embedded zero-tree wavelets (EZW [Shapiro 1993]) or its improved form, the set partitioning in hierarchical trees (SPIHT [Said 1996]). These structures enable to effectively remove the spatial redundancy across multiresolution scales. The additional advantage is the provision of fine scalability. The method of Inoue [Inoue 1998] exploits zerotree structure by replacing the insignificant coefficients with the addition/subtraction of small values. Uccheddu *et al.* adopt a wavelet framework in their blind watermarking scheme for 3D models under the assumption that the host meshes are semi-regular, thus paving the way for a wavelet decomposition and embedding of the watermark at a suitable resolution level [Uccheddu 2004]. For the sake of robust-

ness the host mesh is normalized by a Principal Component Analysis (PCA) before embedding. Watermark detection is accomplished by computing the correlation between the watermark signal and the *to-be-inspected* mesh. Yu *et al.* propose a robust 3D graphical model watermarking scheme for triangle meshes that embeds watermark information by perturbing the distance between the vertices of the model to the center of the model [Yu 2003]. With robustness and perceptual transparency in focus, the approach distributes information corresponding to a bit of the watermark over the entire model. The strength of the embedded watermark signal is adaptive with respect to the local geometry of the model. A method adopts Guskov's multiresolution signal processing method for meshes and uses his 3D non-uniform relaxation operator to construct a Burt-Adelson pyramid for the mesh, and then watermark information is embedded into a suitable coarser mesh [Yin 2001]. The algorithm is integrable with the multiresolution mesh processing toolbox and watermark detection requires registration and resampling to bring the attacked mesh model back into its original location, orientation, scale, topology and resolution level.

Besides above there may be methods involving specialized wavelets. Vatsa *et al.* present a 3-level redundant DWT (RDWT) biometric watermarking algorithm to embed the voice biometric Mel Frequency Cepstral (MFC) coefficients in a color face image of the same individual for increased robustness, security and accuracy [Vatsa 2009]. Green channel is not used and after transforming the red and blue channels, watermarking is carried out followed by the inverse transform. Phase congruency model is used to compute the embedding locations which preserves the facial features from being watermarked and ensures that the face recognition accuracy is not compromised. The proposed watermarking algorithm uses adaptive user-specific watermarking parameters for improved performance. Yen and Tsai put forward an algorithm based on Haar DWT for the gray scale watermark by proposing visual cryptographic approach to generate two random shares of a watermark: one is embedded into the cover-image, another one is kept as a secret key for the watermark extraction later [Yen 2008].

### 4.7.2 Quantization-based methods

The authentication scheme described in [Piva 2005] embeds an image digest in a subset of the subbands from the DWT domain. 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. A JPEG2000-based image authentication method employs extended scalar quantization and hashing for the protection of all the coefficients of the wavelet decomposition [Schlauweg 2006]. The process involves feature

extraction by wavelets to result in digital signature which, after encryption and error correction coding, is embedded as a removable watermark using the well-known quantization index modulation technique called dither modulation. The embedded watermark information is removable during the decompression process which is important for the improved image quality in the context of visualization. Traditionally, correlation analysis has been an integral part of the SS methods reported in various works - the principal difference being in the manner they ascertain the threshold for decoding. Based on the significant difference of wavelet coefficient quantization, a blind algorithm groups every seven non-overlap wavelet coefficients of the host image into a block [Lin 2008]. The largest two coefficients in a given block are referred to as significant coefficients and their difference as significant difference. The local maximum wavelet coefficient in a block is quantized by comparing the significant difference value in a block with the average significant difference value in all blocks. The maximum wavelet coefficients are so quantized that their significant difference between watermark bit 0 and watermark bit 1 exhibits a large energy difference which can be used for watermark extraction. During the extraction, an adaptive threshold value is designed to extract the watermark from the watermarked image under different attacks. To determine the watermark bit, the adaptive threshold value is compared to the block-quantized significant difference. Jin *et al.* employ modulo arithmetic to constrain the noise resulted from the blind embedding into the quantized DWT coefficients directly. Ohyama *et al.* extract a least significant bit plane (LSB)of the quantized wavelet coefficients of the Y color component in a reversible way. They then embed the secret data and a JBIG2 bit-stream of a part of the LSB plane as well as the bit-depth of the quantized coefficients on some code-blocks [Ohyama 2008]. There are methods that employ quantization index modulation (QIM). The idea is to quantize the host signal with a quantizer indexed by the message. i.e. if $S$ is the embedded signal, M the message, and C the cover or host signal, then $S(C, M) = Q_M(C)$. The embedded signal should then be composed only of values in the set of quantizer outputs [Sullivan 2004]. In the method of Ishida *et al.*, the QIM-JPEG2000 steganography, QIM is exploited with two different quantizers (one for embedding a '0' and other for a '1') to embed bit at the quantization step of DWT coefficients under the assumption that the probabilities of '0' and '1' are same in the message [Ishida 2008]. Based on the compression ratio Li and Zhang propose an adaptive watermarking with the strength of watermark being proportional to the compression ratio to enable the embedded watermark survive the following code-stream rate allocation procedure without degrading the image quality [Li 2003].

### 4.7.3 Embedding in the compressed bitstream

These methods usually involve partial or complete roll back of some coding steps, lazy mode coding. The blind scheme proposed in [Su 2001] integrates data hiding with the embedded block coding with optimized truncation (EBCOT) and embed data during the formation of compressed bit stream. The method of Su and Kuo

employs lazy coding to speed up the encoding process by skipping the 4 lowest bit planes during arithmetical encoding [Su 2003]. The authors maintain a software by the name stegoJasper, as reported in [Kharrazi 2006] in which the bits are modified in function to their contribution in the reconstructed image at the decoder side, i.e. bits with least level of contributions are modified first. With this backward embedding approach they try to minimize the embedding artifact on the final embedded image. A similar method rolls back the JPEG2000 encoding process until the dequantization stage [Noda 2003]. The method relies on the fact that the data has already passed the rate controller during the first encoding and an aspired bitrate has already been established. Hence the second rate control should not be able to remove further information, so the additional information can be embedded after the quantization stage and the manipulated image data are again processed by the remaining parts of the JPEG2000 pipeline. To ensure the fidelity of the embedded data to further processing, the target bitrate may be set at a lower value for initial processing and set to the desired value for the second and final run. The technique is applicable during encoding as well as to already encoded JPEG2000 bit streams. One particular technique embeds watermark in the JPEG2000 pipeline after the stages of quantization and region of interest (ROI) scaling but before the entropy coding [Meerwald 2001b]. A window sliding approach is adopted for embedding and for the sake of reliability the finest resolution subbands are avoided while the lowest frequencies carry higher payload.

## 4.8 Summary

We discussed, in this chapter, the what, when, how and where of information hiding. Before classifying the field, we pointed out the four important factors of hiding capacity, robustness, imperceptibility and tamper resistance. Keeping in view the perceptual and robustness models of information hiding we discussed in some detail one representative embedding technique for each, i.e LSB and SS, respectively. But the main thrust was on the wavelet domain embedding with a focus on JPEG2000. In this context wavelet embedding methods both before and after the advent of JPEG2000 were discussed. This chapter concludes Part I and hence serves as the theoretical foundation for the next part of practical work.

## Conclusion of Part I

*Dans cette première partie du manuscrit, nous avons développé les bases théoriques nécessaires pour la compréhension de notre travail qui est présenté dans la seconde partie de cette thèse. En raison de la diversité des concepts, cette première partie a été divisée en trois chapitres. Le Chapitre 2 a eu pour objectif de présenter le codeur JPEG2000. A partir du concept des ondelettes, nous nous sommes concentrés sur la TOD. Dans ce contexte, des algorithmes classiques de TOD ont été décrits. Après avoir présenté les transitions entre les codeurs JPEG et JPEG2000, nous avons détaillé le schéma général de JPEG2000. Pour terminer ce chapitre, les aspects évolutivité et taux de contrôle ont été présentés. Dans le Chapitre 3, nous avons présenté un aperçu des possibles représentations 3D. Après une description des méthodes de visualisation, basées une surface ou sur un volume, ce chapitre donne une attention particulière aux représentations basées surface pour la visualisation 3D de modèles numériques de terrains et de visages. Dans le Chapitre 4, nous avons introduit et présenté les différents aspects des méthodes d'insertion des données cachées, en tenant compte des contraintes et en proposant une catégorisation des techniques d'insertion et des métriques nécessaires à l'évaluation. Dans ce chapitre, nous nous sommes ensuite focalisés sur l'art de cacher des données dans le domaine des ondelettes et en particulier dans le codeur JPEG2000. Nous avons montré les différents endroits où l'on peut interrompre le codeur afin d'intégrer des données.*

In this first part of the manuscript, we developed the necessary theoretical foundation for the comprehension of our work to be presented in the next part. Due to the diversity of concepts, the first part has been divided into three Chapters. Chapter 2 aims to present the JPEG2000 codec. Starting from the concept of wavelets, we focused on the DWT. In this context, some familiar DWT algorithms were presented. After introducing the transitions between JPEG and JPEG2000 encoders, we explained the general schema of the codec. In the end, the aspects of scalability and rate control were presented. In Chapter 3 we outlined a brief overview of 3D representation. After a description of visualization, based on surface and volume, this chapter gives special attention to surface-based representations for the 3D visualization of digital terrain models and faces. In Chapter 4 we introduced and presented the various aspects of information hiding, including the constraints and proposing categorization of embedding techniques and the metrics needed for evaluation. In this chapter, we then focused on the art of hiding data in the wavelet domain and in a particular in the JPEG2000 encoder. We showed different stages where one can interrupt the encoder in order to embed the data.

# Part II

# CONTRIBUTIONS

*Comme décrit dans le Chapitre 3, la visualisation 3D a souvent besoin de plus d'un seul fichier. En général, une carte du profondeur et une texture sont primordiaux. Le niveau de résolution et la qualité des donnés s'améliorent de jour en jour avec l'avènement de nouveaux capteurs toujours plus efficaces. Toutes ces améliorations ont un coût dans la mesure où le traitement, la mémoire et la transmission par réseau sont concernés. Dans un environnement client/serveur, la qualité de visualisation dépend de la charge du réseau, du débit, du serveur distant et du taux de transfert des données. Pour faire face à toutes ces contraintes, il est impensable d'envisager de mettre en place une stratégie sans se concentrer sur la compression des données. Mais en plus de la stratégie de compression, d'autres aspects sont également à prendre en compte du fait que les capacités de ressources et de besoins des clients sont différentes. La large gamme de clients en terme de capacité nécessite donc une compression de données devant être modulable hiérarchiquement afin de satisfaire tout type de clients. La norme JPEG2000 sert cet objectif pour les images (et donc pour les textures) car elle utilise une transformée en ondelettes et les ondelettes ont la propriété de multirésolution. Cependant, pour de la visualisation 3D à distance adaptée aux capacités des clients, la mise en place de protocoles est nécessaire afin de tenter d'unifier les données disparates 3D. Dans le Chapitre 5, cette unification est effectuée grâce à l'insertion synchrone des coefficients transformés de la carte de profondeur en utilisant une méthode d'insertion basée sur la substitution des bits de poids faibles des coefficients de la carte de texture dans le codeur JPEG2000. Les méthodes présentées dans ce chapitre sont imperceptibles, mais peu robustes. Dans le Chapitre 6, suite à une analyse psycho-visuelle du modèle 3D en fonction du point de vue, nous adaptons la synchronisation afin de s'assurer que le niveau d'erreur est tolérable, que ce soit en texture ou en profondeur. Dans la continuité de cette adaptation, nous propopons d'insérer les données uniquement dans un sous-ensemble plutôt que dans la totalité des sous-bandes. Pour des raisons de robustesse, dans cette partie nous avons utilisé une stratégie d'insertion par étalement de spectre qui est à la fois robuste et imperceptible du fait qu'il est possible de retirer la marque après extraction. Une fois cette proposition d'adaptation présentée, dans le Chapitre 7 nous prenons en compte le fait d'avoir une vue plus large en se concentrant sur la jonction de dalles voisines codées séparement avec les méthodes que nous proposons. Ces jonctions sans raccords visibles sont réalisées par lissage orienté dans le domaine des ondelettes pendant le décodage.*

As described in Chapter 3, the 3D visualization needs more than one file: the range data and the texture being the essential. The resolution and quality of the data are improving day by day with the advent of newer more efficient sensors. All these improvements have a cost as far as processing, memory and network requirements are concerned. In a client/server environment the visualization quality depends on the payload, bandwidth, distant server and data transfer rate. To cope with all these constraints, 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. The wide spectrum of clients, in terms of capacity, thus necessitates the data compression to be hierarchically scalable in order to satisfy each type of clients. The JPEG2000 standard serves this purpose for images (and hence for texture) since it is wavelet-oriented and wavelets have the property of multiresolution. Besides, for distant 3D visualization scalable to the client capabilities, the implementation of protocols is necessary to try to unify the disparate 3D data. In Chapter 5 this unification is achieved through synchronous LSB embedding of DWT domain range map coefficients in the DWT domain texture from the JPEG2000 encoder. The methods presented in the chapter are perceptually transparent but not that robust. Later, in Chapter 6, after performing a psycho-visual analysis of the 3D model according to the viewpoint, we adapt the synchronization to ascertain that the level of error is tolerable, both in texture and the depth map. As a follow up of this adaptation we try to embed in a subset, rather than all, of subbands. For the sake of robustness we employ, in that part, a spread spectrum strategy for embedding which is at the same time imperceptible due to its removable nature. Once this proposed adaptation is presented we take a broader view of the problem in Chapter 7 by concentrating on the seamless joining of neighboring tiles, separately encoded by our prosed methods. This seamless tessellation is carried out by DWT domain smoothing during the decoding.

Chapter 5

# Methods with Synchronized Embedding

Contents



## 5.1 Introduction

*Avec l'évolution importante des technologies actuelles, même si la qualité de visualisation 3D devient très élevée, les environnements client/serveur restent très différents en termes de réseau, de calcul et de mémoire. De ce fait, afin de satisfaire*

*chaque client potentiel, il est recommandé de coder les données de façon progressive et unifiée dans un fichier respectant un format standard. Le format JPEG2000 offre l'évolutivité grâce à sa nature multirésolution de la transformée en ondelettes discrètes (TOD). Pour la synchronisation des données, il est possible de s'appuyer sur des techniques d'insertion de données cachées du fait de la taille réduite de la carte de profondeur par rapport à la taille de l'image de texture qui est bien plus volumineuse. Mais cette insertion doit être effectuée de telle manière que le format du fichier JPEG2000 soit conservé. En plus, l'insertion ne doit pas interférer avec l'aspect hiérarchique du codeur JPEG2000. De ce fait, pour chacune des résolutions possibles, la texture et la carte de profondeur correspondant doivent pouvoir être récupérables au niveau du décodeur. Dans ce chapitre, la synchronisation de la carte de profondeur avec la texture correspondante est réalisée avec des stratégies d'insertion de données cachés dans le domaine des ondelettes de façon imperceptible. Afin de conserver une très bonne qualité de visualisation, nous nous appuyons sur une insertion dans les bits de poids faibles. Les méthodes proposées sont aveugles dans le sens que seule une clé secrète et la taille de la carte de profondeur sont nécessaires pour extraire les données de l'image de texture.*

With the evolution of existing technologies, even if the quality of 3D visualization becomes very high, the client/server environments are very diverse in terms of network, computation and memory resources. Therefore, to cater each of the perspective client, it is advisable to encode the data in a scalable way, unified into one standard format file. The JPEG2000 format offers the scalability thanks to the multiresolution nature of its discrete wavelet transform (DWT). For the integration of all the data into one file one can rely on the technique of data hiding due to the smaller size of the depth map file as it can be embedded in the bulky texture image. But this embedding must be carried out in such a way that the JPEG2000 file format is conserved. In addition, the embedding must not interfere with the multiresolution hierarchy of the JPEG2000. As a consequence, for each of the possible resolutions, the corresponding texture and its depth map must be recoverable at the decoder. In this chapter, the synchronized unification of the range data with the corresponding texture, is realized by the application of perceptually transparent DWT domain data hiding strategies. In order to conserve the high quality of visualization we are relying on the LSB based embedding. The proposed methods are blind in the sense that only a secret key, if any, and the size of the range image are needed to extract the data from the texture image.

Our study envisages both types of possible transformations, that is the lossless (Section 5.2) and lossy cases (Section 5.3). As a first step, we tested our methods on 3D visualization of DEM. The results were then applied in the context of 3D face visualization, in Section 5.4, with the case where the difference in sizes of the range data and its texture is small. Concluding remarks are given in Section 5.5.

## 5.2 Synchronized Embedding in the Lossless JPEG2000

We propose over here a least significant bit (LSB) based strategy that would embed the losslessly transformed range data in the corresponding transformed luminance/chrominance component of texture from the lossless JPEG2000 coding pipeline. The method is based on the works presented in [Hayat 2006, Hayat 2007c] that did not involve the JPEG2000 codec, i.e. the DWT transformed DEM was embedded in the transformed texture. An overview of the method is given in Section 5.2.1. Section 5.2.2 presents the transformation of the range data and the texture. The embedding is explained in Section 5.2.3 while the decoding and 3D visualization are elaborated in Section 5.2.4. Detailed qualitative and quantitative results are explained in Section 5.2.5 while conclusions are given in Section 5.2.6.

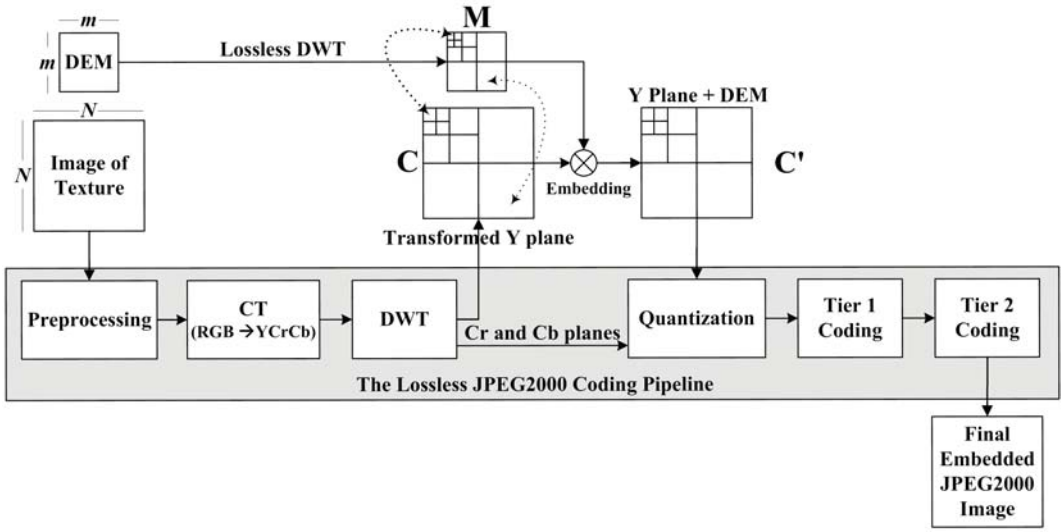



Figure 5.1: Embedding transformed DEM coefficients into the transformed coefficients of each Y texture subband from the lossless JPEG2000 encoder.

### 5.2.1 Overview

In this section 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 5.1.

### 5.2.2 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 may be sufficient (altitude between $-8192$ $m$ and $+8192$ $m$), but this may not be the case for DWT coefficients and calculations may lead to overflowed results which makes us allot two additional bits, i.e. 16 bits to represent each coefficient in the DWT domain in order to maintain precision. 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. 5.1, 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 [Mallat 1998] that employs the JPEG2000 supported reversible Daubechies (5/3) [Daubechies 1998].

### 5.2.3 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. 5.1. 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.

In the Fig. 5.2 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 $u, v \in \{0, \ldots, m-1\}$. The $N \times N$ transformed Y plane of texture is consequently partitioned into $m^2$ blocks $B_{u,v}$, with $u, v \in \{0, \ldots, m-1\}$. Each of $B_{u,v}$ is composed of $r^2$ $(= N^2/m^2)$ coefficients $t_{i,j}$, with $i, j \in \{0, \ldots, r-1\}$. Let $b_{15} \ldots b_0$ and $\beta_7 \ldots \beta_0$ 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 $\beta_7\beta_6\beta_5.....\beta_1\beta_0$ that would yield $t'_{i,j} = \beta_7\beta_6\beta_5.....\beta_1 b_k$ 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

**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: **initialize** a PRNG with a seed $K$
6: **for** each subband of $C$ and its counterpart in $M$ **do**
7: **partition** the $C$ subband into square blocks in relation to the $M$ subband
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: **label** the block-wise marked $C$ as $C'$
14: **end**

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.

### 5.2.4 Extraction and reconstruction

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 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 $3(L - L') + 1$ 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 $3L'$ 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:

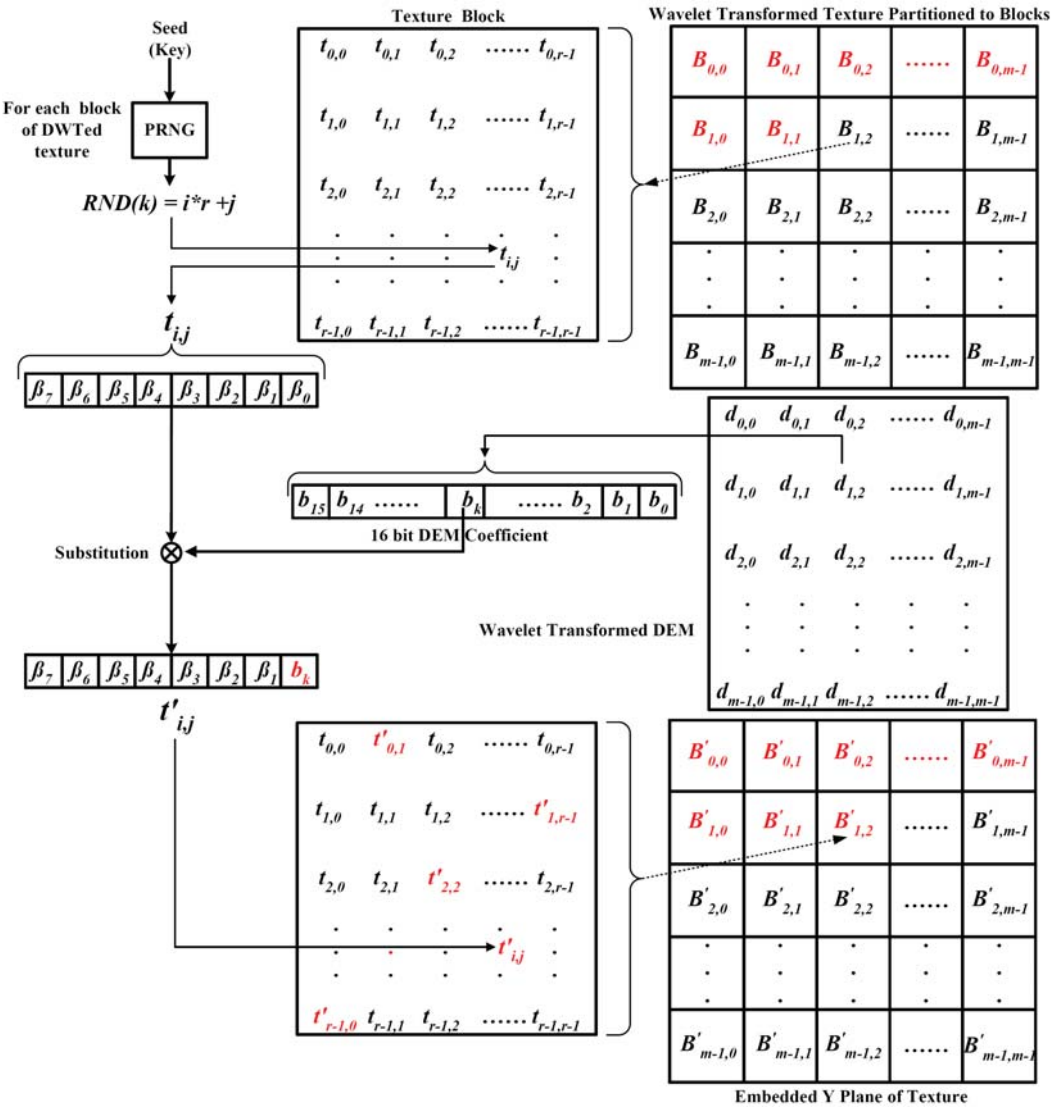



Figure 5.2: Embedding a subband of the transformed DEM coefficient in a block from a subband of the transformed texture.

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. 5.3.

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

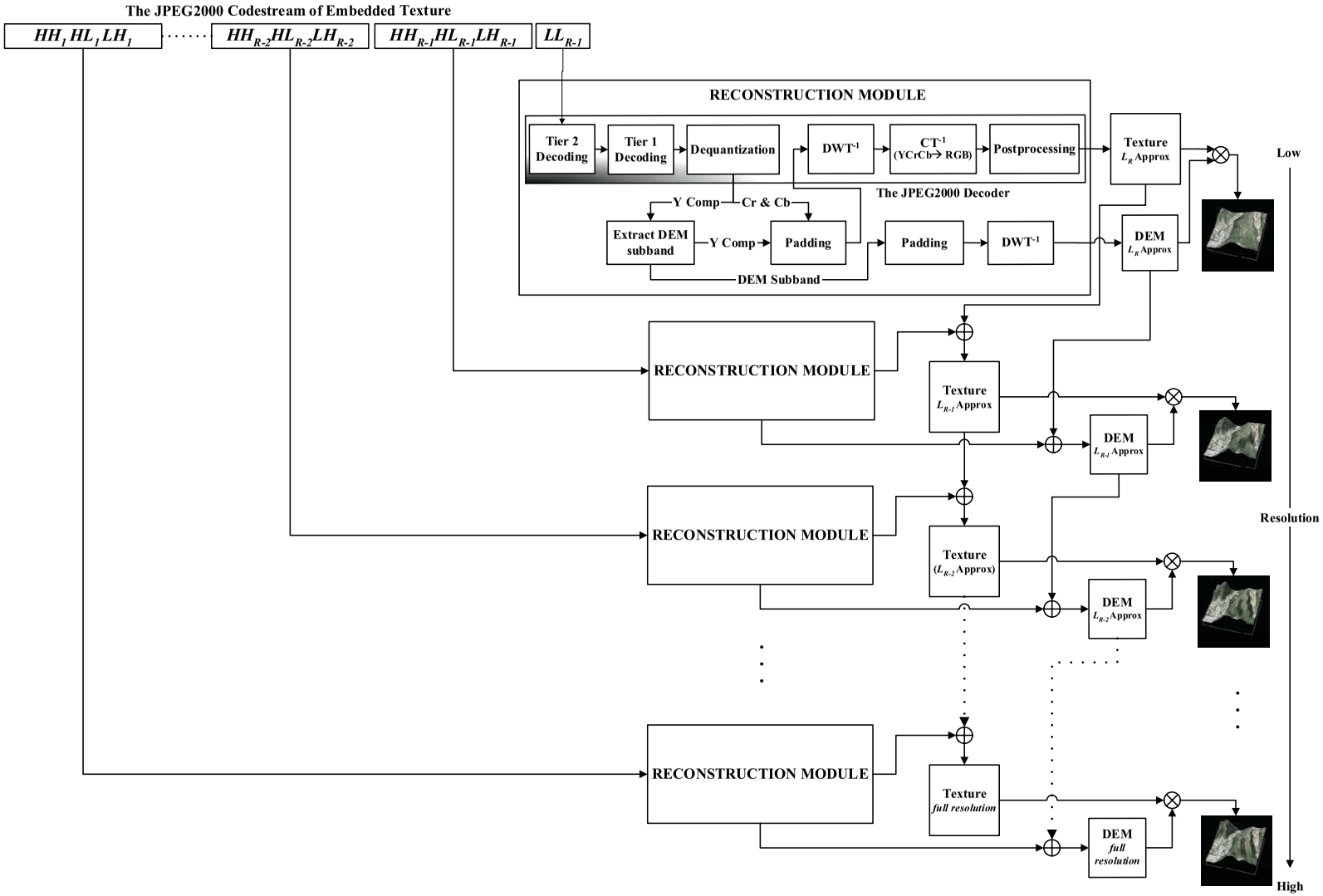



Figure 5.3: Decoding, data extraction and scalable 3D reconstruction and visualization.

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. 5.3) 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.

### 5.2.5 Experimental results

This section has two parts. In Section 5.2.5.1 we analyze the application of our method when applied to an example altitude map of $64 \times 64$ coefficients (Fig. 5.4.a) associated with a $3200 \times 3200$ pixel texture (Fig. 5.4.b). For the purpose of comparison a $128 \times 128$ pixel detail of the above mentioned image of texture is presented in Fig. 5.4.c. Section 5.2.5.2 is concerned with the 3D visualization simulation with respect to another example that is to be demonstrated at high and low bitrates.

(a) (b) (c)

Figure 5.4: Example images; a) original altitude map, b) original texture map, c) a part of texture magnified.

#### 5.2.5.1 Analysis of the method as applied to a real world example

For the example illustrated in Fig. 5.4, each coefficient of the altitude is coded with 2 bytes implying the embedding factor of 1 coefficient per $50 \times 50$ pixels of texture. The application of a level 1 lossless wavelet transformation to the DEM results in Fig. 5.5.a. The corresponding transformed Y-plane of the texture (Fig. 5.5.b) is extracted *in situ* from the JPEG2000 encoding process of the texture image. The corresponding set of images for level 3 is illustrated in Fig. 5.5.c and 5.5.d. Embedding

(a) (b) (c) (d)

Figure 5.5: DWT at level $L$: a) altitude ($L = 1$), b) Y plane of texture ($L = 1$), c) altitude ($L = 3$), d) Y plane of texture ($L = 3$).

of the transformed DEM in the transformed texture takes place according to the policy explained in Section 5.2.3. 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. 5.5.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 altered 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 $6.4 \times 10^{-3}$altitude coefficients per pixel since only one 16 bit DEM coefficient is embedded per $50 \times 50$ block of coefficients of the Y component of texture.

Fig. 5.6.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 resulted in Fig. 5.6.b. When the above process of reconstruction is applied to the level 1 lowest subband of the embedded texture image Fig. 5.6.c is obtained. For comparison with the original example, the $128 \times 128$ pixel magnified part is shown in Fig. 5.6.d. The same sequence of four figures, i.e. a) to d), is

(a) (b) (c) (d)

Figure 5.6: Level 1 approximate reconstruction: a) the extracted DEM, b) difference of the extracted with the original DEM, c) the texture, d) the magnified texture.

(a) (b) (c) (d)

Figure 5.7: Level 2 approximate reconstruction: a) the extracted DEM, b) difference of the extracted with the original DEM, c) the texture, d) the magnified texture.

(a) (b) (c) (d)

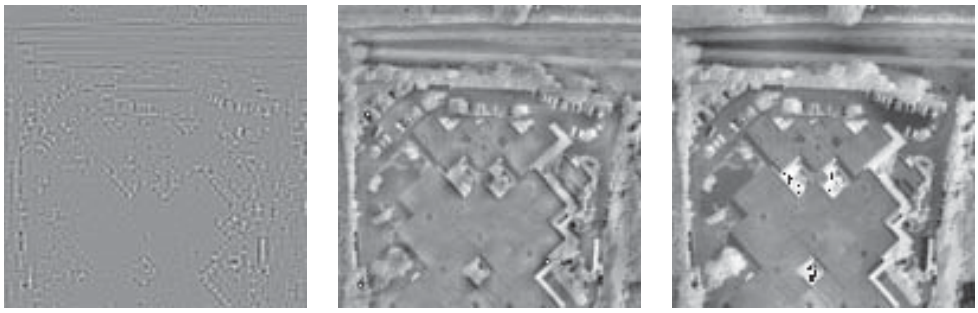
Figure 5.8: Level 3 approximate reconstruction: a) the extracted DEM, b) difference of the extracted with the original DEM, c) the texture, d) the magnified texture.

(a) (b) (c) (d)

Figure 5.9: Level 4 approximate reconstruction: a) the extracted DEM, b) difference of the extracted with the original DEM, c) the texture, d) the magnified texture.

(a) (b) (c) (d)

Figure 5.10: Level 5 approximate reconstruction: a) extracted DEM, b) difference of the extracted with the original DEM, c) the texture, d) the magnified texture.



(a) (b) (c)

Figure 5.11: Difference between the original luminance and reconstructed texture (magnified according to Fig. 5.4.c): a) level 1, b) level 3, c) level 5.

maintained in each of the Fig. 5.7– 5.10 which corresponds to the reconstruction from levels 2 to 5 lowest subbands, respectively. For subjective analysis, one can visually compare the part *d* of Fig. 5.4, with the same of each of Fig. 5.6 – 5.10. 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. 5.11 in the form of difference images in terms of the magnified part corresponding to Fig. 5.4.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 compression rate. The results are summarized in Table 5.1 for the DEM and Table 5.2 for the texture. To elaborate further, MSE has been plotted as a function of bitrate for the texture (Fig. 5.12.a.) and as a function of bits per coefficient for DEM (Fig. 5.12.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. The effectiveness of our method is revealed by the fact that even for a compression rate as low as 0.26 *bpp* (corresponds to level 3 approximation image of texture) one observes a $PSNR$ of

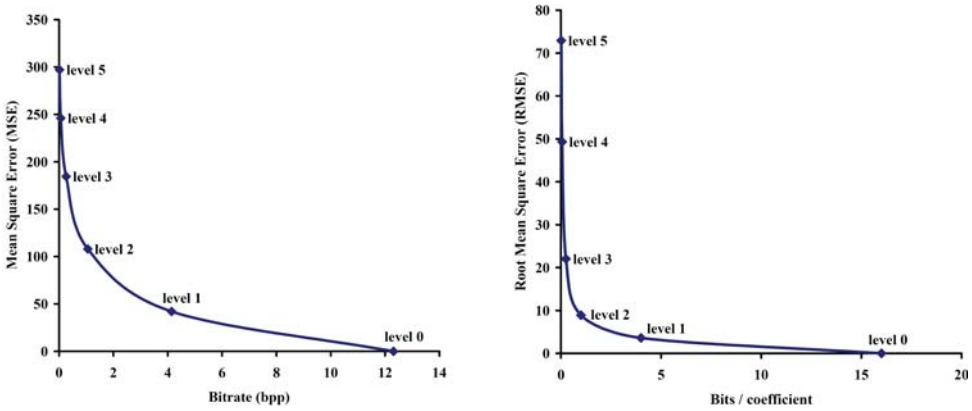
25.47 corresponding to a $RMSE$ of 22.02 $m$. Given the fact that being computed from a small fraction of initial data, this error is tolerable if the observation is made from a very high altitude.

| 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 (m) | 72.92 | 49.33 | 22.02 | 8.90 | 3.60 | 0 |
| PSNR (dB) | 10.87 | 14.27 | 21.27 | 29.14 | 37 | $\infty$ |

Table 5.1: Results obtained for DEM after the extraction and reconstruction as a function of the used data.

| 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 |
| Bit Rate (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 (dB) | 23.40 | 24.22 | 25.47 | 27.80 | 31.9 | 68.53 |

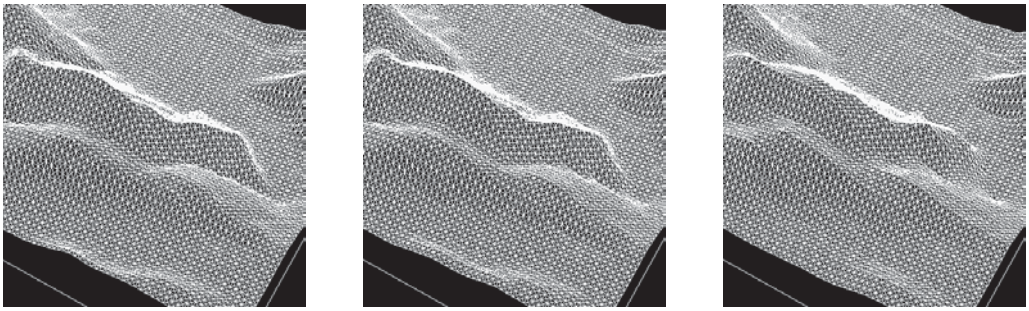
Table 5.2: Results obtained after the extraction and reconstruction of the texture as a function of the used data in the form of embedded texture.



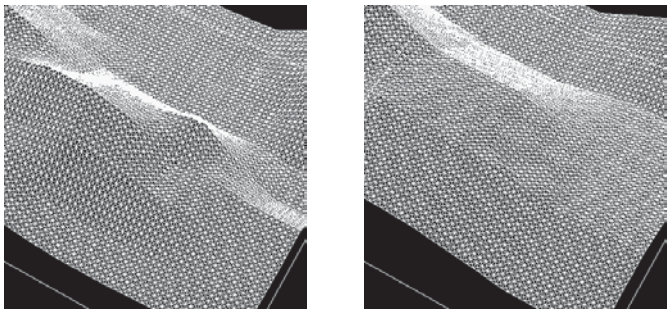

(a) Texture (b) DEM

Figure 5.12: Graphical display of variation in quality as a function of bitrate.

The reconstructed 3D surfaces drawn from the various approximation DEM's are shown in Fig. 5.13. These surfaces when overlaid by their corresponding textures effect 3D visualizations (Fig. 5.14). One can compare the final result between a 3D visualization with all the data (Fig. 5.14.a), after being subjected to our method, and a visualization with the level 1, 2, 3 and 4 (Fig. 5.14.b–e), corresponding to 25%, 6.25%, 1.56% and 0.39% of the transmitted coefficients, respectively.



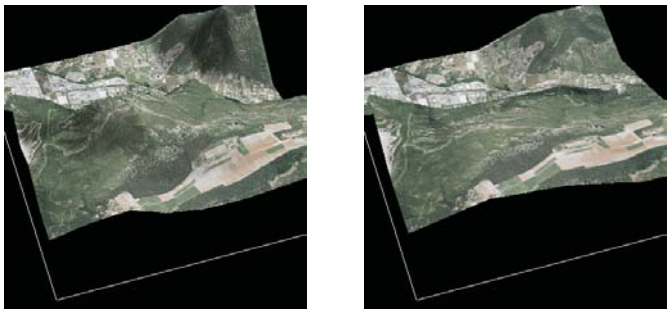
(a) Level 0(all the data) (b) Level 1 (c) Level 2



(d) Level 3 (e) Level 4

Figure 5.13: 3D surface representation using the extracted DEM's and obtaining their approximations at various levels.

(a) Level 0(all the data) (b) Level 1 (c) Level 2



(d) Level 3 (e) Level 4

Figure 5.14: 3D visualization with the images of approximations at various levels.

#### 5.2.5.2 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. 5.15. Let the terrain is decomposed into tiles for a real time visualization with a tile having the DEM ($64 \times 64$) from Fig. 5.15.a and its corresponding $3200 \times 3200$ aerial image (Fig. 5.15.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.

(a)  (b) 

Figure 5.15: Example utilized in the simulation: a) the altitude map, b) the texture.

Five different snapshots from the 3D visualization of our example are illustrated Fig. 5.16 for level 0 (corresponding to a bitrate of 240 Mbps) and Fig. 5.17 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. 5.16 and 640 kbps in Fig. 5.17) at a given interval. The difference is obvious but not glaring given the fact that the data of Fig. 5.17 corresponds to only 0.39% of the number of coefficients of the data of Fig. 5.16. This lower than expected degradation in quality motivate us to have an effective scalable visualization. For example a $3200 \times 3200$ 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 arround 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.

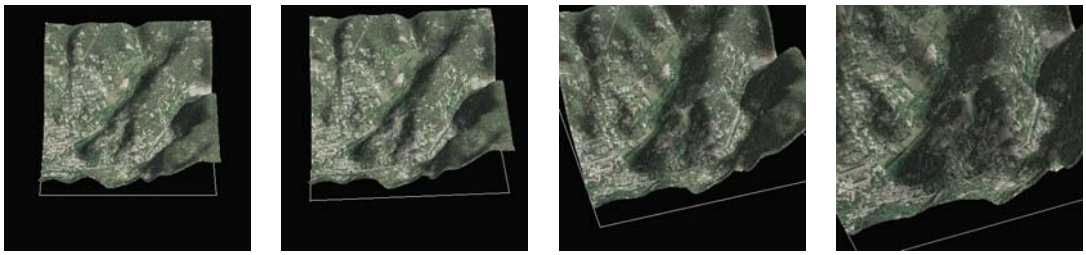


Figure 5.16: Snapshots of the 3D visualization simulation based on **level 0 (for** 240 **Mbps)** approximations after extraction and reconstruction.

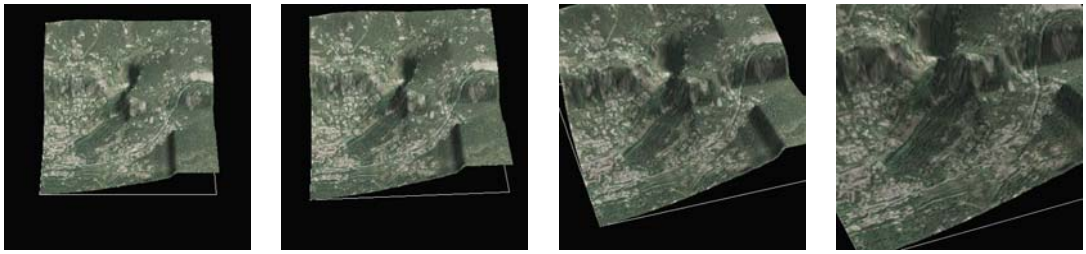


Figure 5.17: Snapshots of the 3D visualization simulation based on **level 4 (for** 640 **kbps)** approximations after extraction and reconstruction.

For the level-4 approximation, a closer examination of Fig. 5.17 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.

### 5.2.6 Synopsis

The results shown in the case of our examples are witness to the fact that 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. The interesting outcomes of the method presented in this section led to at least three publications [Hayat 2007a, Hayat 2007b, Hayat 2008b].

## 5.3 Synchronized Embedding in the Lossy JPEG2000

In this section a blind LSB-based data hiding method is proposed for the synchronous unification of this disparate data whereby the lossless discrete wavelet transformed (DWTed) range coefficients are embedded in the tier-1 coded quantized and DWTed Y/Cr/Cb component of the texture image from the lossy JPEG2000 pipeline. The strategy is synchronous in the sense that the level of transform of both the carrier and the message is the same and low energy subbands carry the lower subband coefficients of DWTed DEM and vice versa. Hence from any available subset subbands one can recover the corresponding subset for the range map. The multiresolution nature of DWT fulfills our requirement of scalability. In a typical lossy JPEG2000 pipeline the two key steps with potential loss of data are quantization and tier-1 (T1) coding. These two steps come after the irreversible color and DWT (9/7) transforms. For the lossless case the quantization is reversible and that is why in the method, presented in the previous section, the intervention came immediately after the DWT(5/3) step. For the lossy case we have changed the strategy since quantization is now inevitable. What we did is to interrupt the flow after the T1 coding as shown in Fig. 5.18.

### 5.3.1 Overview

In T1 coding, which is the first of the two entropy coding steps of JPEG2000, the quantizer indices for each subband are partitioned into rectangular code blocks with its nominal dimensions being dyadic and their product not exceeding 4096. The partitioned code blocks are coded independently using the bit-plane coder thus generating a sequence of symbols with some or all of these may be entropy coded[1]. Due to this independent encoding of code blocks, the correspondence between the lossless DWTed DEM and lossy DWTed Y plane of texture is maintainable. The T1 coded symbols from a given block vary in energy and the low index symbols are more energetic than the higher index ones. What we do is to use the least energetic of these symbols, from the tail of the stream for each code block, for LSB embedding implying non-random allocation. There is, however one problem in that the T1 coded symbols have smaller wordsize resulting in smaller embedding capacity and higher rate of distortion in quality as a result of embedding. This policy is not, however, advised in the lossless case since wordsizes of the coefficients are longer at the earlier steps thus leading to lesser distortions as result of embedding. Hence one can embed immediately after the DWT step at the earliest.

### 5.3.2 The Data hiding step

From a $N \times N$ pixel texture image and the corresponding map of $m^2$ altitudes, we deduce the embedding factor $E = m^2/N^2$ coefficients per pixel. The image of texture will, therefore, have to be divided into square blocks of size equal to

[1] http://www.ece.uvic.ca/∼mdadams/jasper/

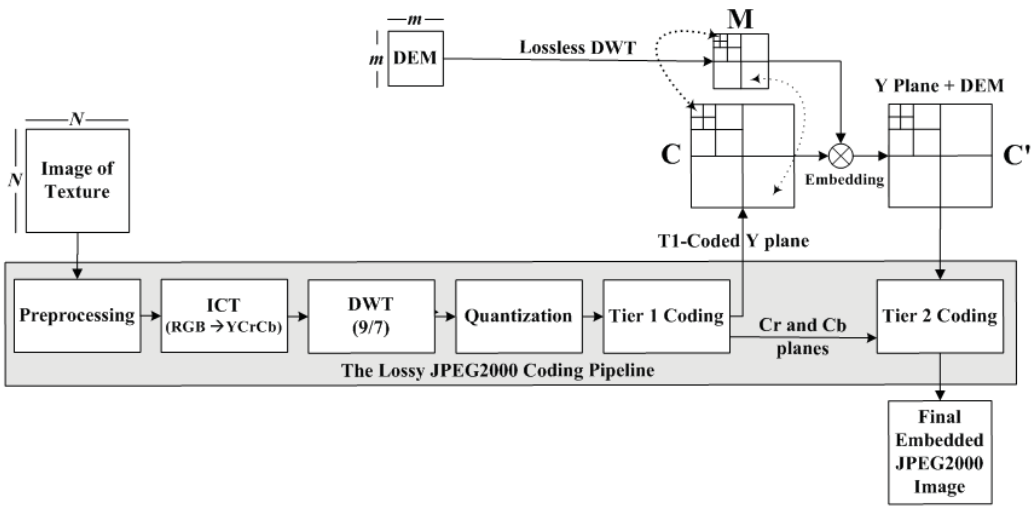



Figure 5.18: The encoding procedure.

$\lceil 1/E \rceil$ pixels and every such block would hide one altitude coefficient. In the first place, the T1-coded quantized level $L$ DWTed Y component (say carrier $C$) of the texture image is extracted from the lossy JPEG2000 pipeline that employs the irreversible Daubechies (9/7) [Daubechies 1998]. A discrete wavelet transformation (DWT) is separately applied to the DEM coefficients at the same level $L$ to get the message $M$, as illustrated in Fig. 5.18. The implementation of the discrete wavelet transformation (DWT) of the DEM is based on the lifting method [Mallat 1998] that employs the JPEG2000 supported reversible Daubechies (5/3) [Daubechies 1998]. It has been ensured that both $M$ and $C$ are at the same level of transformation and embedding is according to the correspondence of subbands, i.e. lower frequency subbands of cover carry the lower subband coefficients of message as illustrated in Fig. 5.18. The basics are the same as explained in Section 5.2.3. The data embedding is, however, carried out by modifying the LSB's of a certain number of coefficients of the T1-coded lowest energy symbols resulting in $C'$. The final encoded image thus carries the DEM coefficients hidden in some T1-coded symbols from the Y component data.

### 5.3.3 The extraction/reconstruction step

The above coded image can be utilized like any other JPEG2000 image and sent across any communication channel. The blind decoding is the reverse of the above process. Just before the T1 stage of the JPEG2000 decoder the DEM can be blindly extracted using the above mentioned partitioning scheme. All the DEM bits are LSBs of the lowest energy symbols for each of the given code block from the T1-coded quantized DWTed Y component. One advantage of the method is in the fact that the DEM and 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 $3D$ visualization even with fewer than original resolution layers as a result of partial or delayed data transfer. The method thus enables to effect visualization from a fraction of data in the form of the lowest subband, of a particular resolution level. It is always possible to stuff 0's for the higher bands. The idea is to have a 3D visualization utilizing $3(L-L')+1$ parts out of the initial $3L+1$ parts ($L' \leq L$), by stuffing the rest of $3L'$ parts with 0's. The reconstruction process is therefore the same as already illustrated in Fig 5.3 with a little modification as now one has to intervene earlier, i.e. before T1-decoding.

### 5.3.4 Experimental results

We have applied the method to a $2048 \times 2048$ pixel example[2] texture image (Fig. 5.19.b) and its corresponding DEM of $64 \times 64$, 16 bit altitudes implying one altitude per $32 \times 32$ pixel texture block. By adding a suitable header to the DEM it is converted to a gray-scale image (Fig. 5.19.a) with whiter parts of the image representing high and black parts representing the low altitudes. We chose to subject the texture to lossy JPEG2000 encoding at $L = 5$. For the sake of comparison our reference will be the lossless method described in section 5.2 wherein the level $L$ transformed DEM coefficients were embedded in the corresponding Y plane texture coefficients immediately after the level $L$ DWT step of the lossless JPEG2000 pipeline using a pseudorandom number generator (PRNG) for allocation. The reason for not using the same point of intervention for lossless version has already been explained elsewhere in this section.

(a) (b) (c)

Figure 5.19: Example data : a) DEM as image, b) texture, c) a $128 \times 128$ texture detail.

Level $L$ coded embedded texture image can give $L+1$ possible approximation images upon decoding that pertains to resolutions numbered $0, 1, \ldots, L$ in descending order of quality. An $L$ approximate image is constructed from data amounting to $100/4^L\%$ of the total coefficients with the rest of coefficients (missing) being replaced by 0's before inverse DWT. This is true of all the components. Thus, a level 5 image of approximation is constructed only with 0.1% of the coefficients. Since the

[2]provided by IGN France (http://www.ign.fr/)

data hiding method is synchronous, the level of approximation of both the texure and DEM for a given chunk of available coefficients is the same. Fig. 5.20 shows the extracted/reconstructed texture/DEM pairs by stuffing 0's for higher subands of the level $1, 3$ and 5 lowest subbands and inverse DWTing the result to get the images of approximation. One observes that with a naked eye, degradation in the quality of DEM is more as compared to the texture.

A quantitative comparison can be done in terms of the images of difference by observing the measures, like MSE or PSNR, as a function of compression rate. It must be noted, that irrespective of the method used, the results are the same as far as the quality of the reconstructed DEM is concerned. Fig. 5.21.a depicts the trend of these results when $\sqrt{MSE}$ is plotted as a function of bits per coefficient for DEM (Fig. 5.21.a). We have plotted $\sqrt{MSE}$ against bits per coefficient due to the fact that rather than the whole JPEG2000 encoding only DWT has been applied to the DEM coefficients. Since DEM was losselessly DWTed, the resultant quality is the same irrespective of whether the cover Y texture transformed using lossy or lossless DWT before embedding. Fig. 5.21.b-d compares the two methods by taking into account measures like PSNR, MSE and bitrate. By all accounts the proposed method fare better, the most interesting being the curves in Fig. 5.21.b which compares the two methods in terms of texture quality with MSE plotted against bitrate (bpp). It is pertinent to note that for the same bitrate one gets a better quality texture image with the proposed method. For example, for level 1 extracted and reconstructed texture image, with the previous method one got a PSNR of $31.3dB$ and that too with a compression ratio of 5.71. With the proposed method the results were much better and the PSNR was $31.6dB$ and compression ratio of 7.19. The 3D simulation of the two cases can be compared by observing Fig. 5.22. The corresponding 3D visualization images of the original example and the rest of the levels of approximation in case of the proposed method are given in Fig. 5.23. It can be seen that even with a very tiny fraction (0.2 bpp) of data one can have a relatively better visualization.

### 5.3.5 Synopsis

The results have been interesting and for a bitrate as low as 0.2 bits per pixel (bpp), an acceptable visualization was realized. We compared the obtained results with those of the previous method that interrupt the lossless JPEG2000 codec immediately after the DWT step and embeds lossless DWTed DEM in the reversibly DWTed Y component of texture. What we did is to use the least energetic of these symbols, from the tail of the stream for each code block, for LSB embedding implying non-random allocation. Thus unlike the previous method no PRNG was used for allocation. A casual look at the results may suggests that the proposed method may be more effective, than the one before, in the sense that for the same bitrate one observed lesser quality loss for respective resolutions. There is, however one problem in that the T1 coded symbols have smaller wordsize resulting in smaller embedding capacity and higher rate of distortion in quality as a result of embedding. That is

(a) Level 0

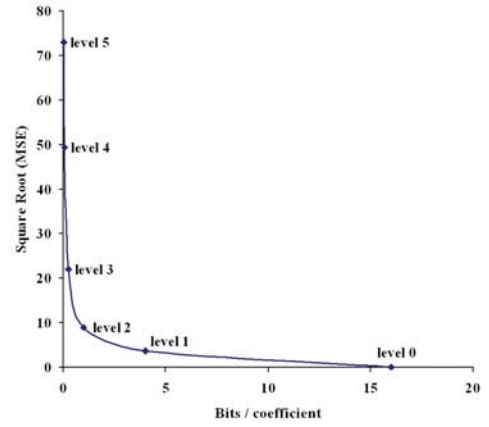
(b) Level 1

(c) Level 2

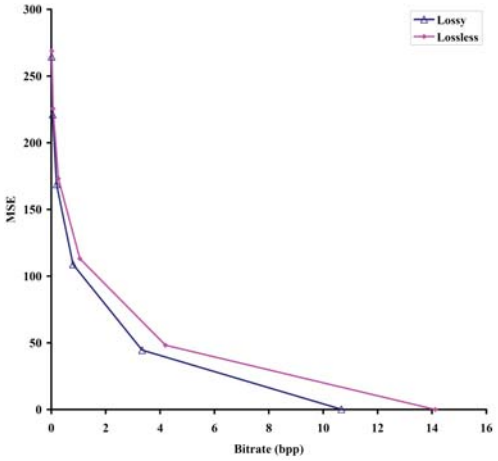
(d) Level 3

(e) Level 4

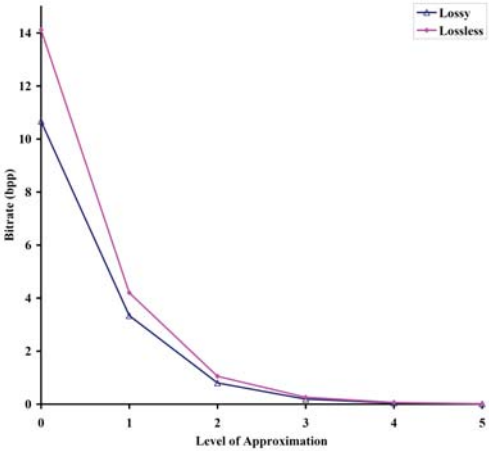
Figure 5.20: Approximation images of DEM/texture pairs.



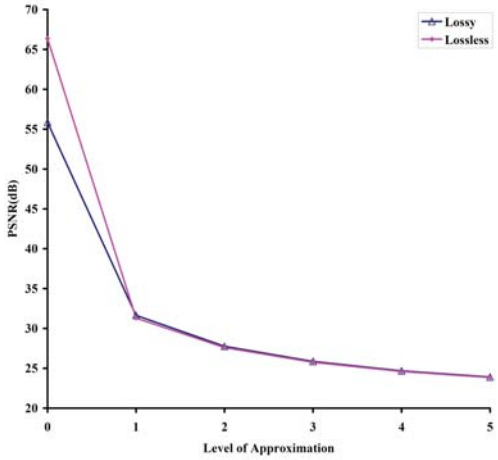

(a) DEM: RMSE vs bits per coefficient




(b) Texture: MSE vs bitrate




(c) Texture: bitrate vs level of approximation



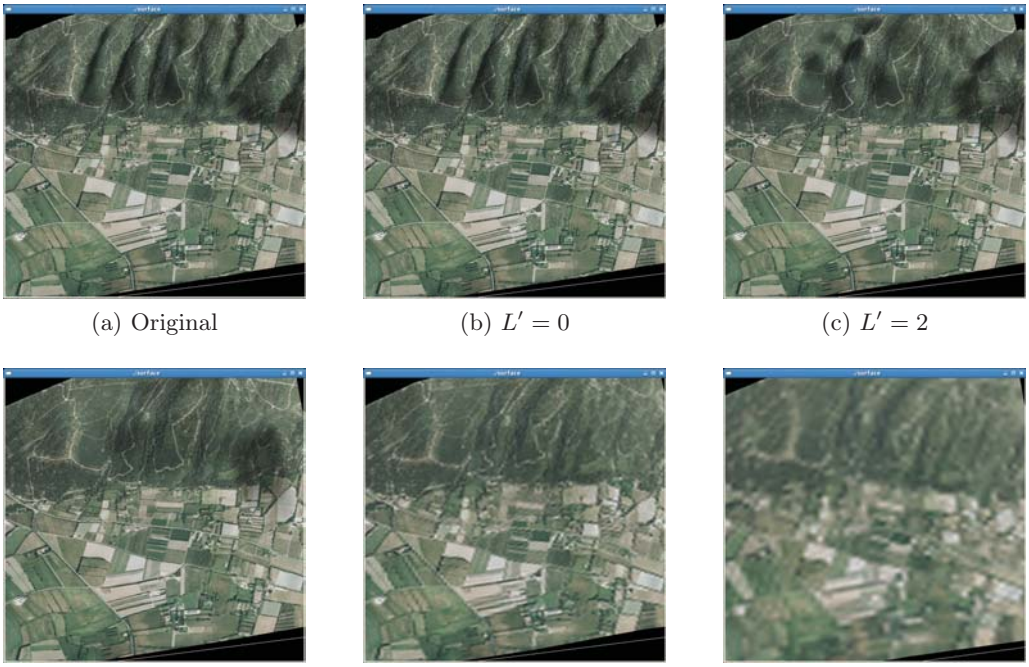

(d) Texture: PSNR vs level of approximation

Figure 5.21: Graphical representation of results.


(a) The proposed method

(b) The method of Section 5.2

Figure 5.22: 3D visualization based on level 1 approximations from two methods.



(a) Original (b) $L' = 0$ (c) $L' = 2$

(d) $L' = 3$ (e) $L' = 4$ (f) $L' = 5$

Figure 5.23: 3D visualization based on level $L'$ approximation images when level 5 DWTed DEM is embedded in level 5 DWTed texture.

why it is not advisable to wait until T1 coding in the lossless case since wordsizes of the coefficients are longer at the earlier steps thus leading to lesser distortions as result of embedding [Hayat 2008a].

## 5.4 Improving the Embedding Process

We now tailor our methods for the cases where the range map and its texture do not differ much as far as their sizes are concerned. To this end a two pronged strategy is being adopted. On one side the data to be embedded is optimized and on the other more than one of the luminance/chrominance texture components are being involved as carriers.

### 5.4.1 Overview

Until now we had the luxury of choosing the potential carrier coefficients from a large population of texture coefficient since the range map was very small as compared to the texture. There are areas, like face visualization, where the size of the depth map closely approaches that of the texture. For the work in perspective, we have chosen to improvise the methods to cater even the worst case scenario, i.e. if both the texture and DEM have identical sizes. This would also help us to have an idea of the embedding capacity for the earlier work as a byproduct. As a case study we are taking a face visualization example.

### 5.4.2 Embedding

For an $N \times N$ pixel facial texture and its corresponding $M \times M$ point depth map ($2.5D$) we propose our data hiding strategy presented in Fig. 5.24. The face texture is subjected to the level-$L$ JPEG2000 encoding in the lossless mode. The encoding process is interrupted after the DWT step to get the three transformed YCrCb face texture components. The corresponding grayscale ($k-1$ bit) depth map is also subjected to level-$L$ lossless DWT in parallel. To ensure the accuracy we expand the word-size for each of the transformed depth map coefficient by one additional bit and represent it in $k$ bits. The DWT domain depth map coefficients are then embedded in the DWT domain YCrCb face texture components while strictly following the spatial correspondence, i.e. low frequency $2.5D$ coefficients in low while higher in higher frequency YCrCb coefficients. This step strictly depends on the ratio, $M : N$, where $M \leq N$. In the worst case, where $M = N$, the $k$ bit transformed $2.5D$ coefficient is equally distributed among the three components and each of the transformed YCrCb texture coefficient carry $\lfloor k \rfloor$ to $\lfloor k \rfloor + 1$ bits. If $M < N$ then, rather than a face texture coefficient, a whole face texture block corresponds to one depth map coefficient and one has the choice of selecting the potential carrier coefficients. This is specially true when $M < N/3$ as one is then compelled to run a pseudo-random generator (PRNG) to select the potential carrier coefficients. In this

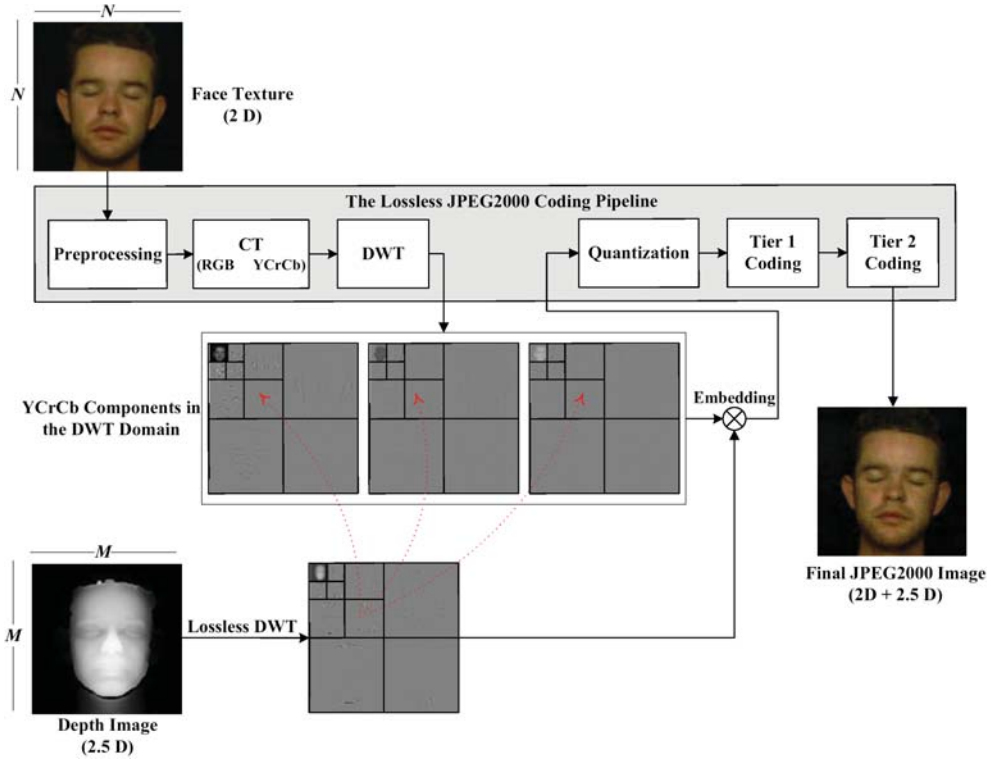



Figure 5.24: Description of the method

scenario one question is important that which of the luminance (Y) or chrominance

(Cr and Cb) be selected for embedding. The answer to this question depends on the extent of compromise between perceptual quality and escalation in the size of final JPEG2000 image. For perceptual transparency, the chrominance planes are the better choice and for better compression the luminance plane is the obvious choice. To keep the method blind, the embedding process involves the substitution of the least significant bit (LSBs) of the carrier coefficient with the bit(s) from the 2.5D coefficient. After embedding, the YCrCb component are re-inserted into the JPEG2000 coding pipeline. The result is a monolithic JPEG2000 format face texture image that has the depth map hidden in it.

### 5.4.3 Improvement

For the work in perspective we have chosen the worst case scenario, i.e. $M = N$. This would also help us to have an idea of the embedding capacity for the earlier work. In the embedding step, a given $k$-bit transformed depth map coefficient is to be substituted into the $[k/3]$ LSBs each of the corresponding Y, Cr and Cb transformed coefficients. To reduce the payload we have optimized our method to some extent. One of the important characteristics of DWT is the high probability of 0 coefficients in higher frequency subbands. Hence one can always use a flag bit to differentiate this case from the rest. In addition, the use of $k^{th}$ additional bit for transform domain coefficients is a bit too much. Thus, for example, for an 8 bit spatial domain $2.5D$ coefficient the initial range of $[-128, 127]$ may not be enough in the DWT domain and needs to be enhanced but not to the extent to warrant a range of $[-256, 255]$. A midway range of $[-192, 192]$ ought to be sufficient. For such a 8-bit scenario one may then have four possibilities for the value of a coefficient viz. zero, normal ($[-128, 127]$), extreme negative ($[-192, -128]$) and extreme positive ($[128, 192]$). Keeping all these possibilities in view, we decided to pre-process the transformed depth coefficient set, before embedding. In our strategy, we keep the first bit exclusively as a flag bit. The next two bits are data cum flag bits and the last six bits are strictly data bits. For a coefficient in the range $[-128, 127]$, the first bit is set to 0, with the rest of eight bits carrying the value of the coefficient, otherwise it is set to 1. For a zero coefficient, the first two bits are set to 1 and thus only 11 is inserted. The absolute difference of an extreme negative coefficient and $-128$ is carried by the last six bits with the first three bits carrying 101. For extreme positives the first three bits have 100 and the rest of six bits have the absolute difference of the coefficient with $+127$. In essence we are to embed either two or nine bits according to the following policy:

- **if** $coeff \in [-128, 127]$ then concatenate coeff to 0 and embed as 9 bits;
- **else if** $coeff = 0$ then embed binary 11;
- **else if** $coeff \in [-192, -128]$ then concatenate $|-128 - coeff|$ to 101 and embed as 9 bits;
- **else** concatenate $(coeff - 128)$ to 100 and embed as 9 bits;

The above coded image can be utilized like any other JPEG2000 image and sent across any communication channel. The blind decoding is the reverse of the above process.

### 5.4.4 Decoding and reconstruction

Just before the inverse DWT stage of the JPEG2000 decoder, the DWT domain depth map can be blindly extracted by reversing the embedding process mentioned above. In the reconstruction phase, by the application of 0-*padding*, one can have $L+1$ different approximation images of facial texture/depth map pair. And this is where one achieve the scalability goal. Our method is based on the fact that it is not necessary that all the data is available for reconstruction. This is one of the main advantages of the method since the depth map and facial texture can both be reconstructed with even a small subset of the transmitted carrier coefficients. The resolution scalability of wavelets and the synchronized character of our method enable a $3D$ visualization even with fewer than original resolution layers as a result of partial or delayed data transfer. The method thus enables 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 0's for the higher bands. The idea is to have a $3D$ visualization utilizing lower frequency subbands at level $L'$, with $L' \leq L$. For the rest of $3L'$ parts one can always pad a 0 for each of their coefficient. The inverse DWT of the 0-stuffed transform components will yield what is known as image of approximation of level $L'$. A level-$L'$ approximate image is the one that is constructed with $(1/4^{L'}) \times 100$ percent of the total coefficients that corresponds to the available lower $3(L-L')+1$ subbands. For example, level-0 approximate image is constructed from all the coefficients and level-2 approximate image is constructed from 6.12% of the count of the initial coefficients. Before being subjected to inverse DWT, data related to depth map must be extracted from the transformed face texture whose size depends both $L$ and $L'$. Thus if $L' = L$ one will always have the entire set of the embedded DEM coefficients since all of them will be extractable. We would have a level 0 approximate final DEM after inverse $DWT$, of the highest possible quality. On the other hand if $L' < L$, one would have to pad 0's for all coefficients of higher $3L'$ subbands of transformed DEM before inverse $DWT$ that would result in a level $L'$-approximate DEM of an inferior quality.

### 5.4.5 Experimental results

We have applied our method to a number of examples from FRAV3D[3] database. One such example is given in Fig. 5.25 that consists of a $120 \times 120$ point 2.5D depth map (Fig. 5.25.a) corresponding to a $120 \times 120$ pixel colored 2D face image given in Fig. 5.25.b. Each point of the 2.5D depth map is coded with 8 bits. The 3D visualization based on the two images is depicted by a view given in Fig. 5.25.c. Lossless DWT is applied in isolation to the depth map at level-3 to get the image

[3] http://www.frav.es/databases/FRAV3d/

given in Fig. 5.26.a. To ensure the accuracy we represent each of the transformed depth map in 9 bits. The corresponding 2D face image is subjected to level-3 lossless JPEG2000 encoding and the process is interrupted just after the DWT step. What we get are the level-3 transformed luminance and chrominance components given in Fig. 5.26b-d. The transformed depth map is embedded in the three components according to the scheme outlined above. The resultant components are reintroduced to the JPEG2000 pipeline at quantization step. The final result is a single JPEG2000 format 2D image.

(a) (b) (c)

Figure 5.25: Original data: a) a $120 \times 120$ depth map (2.5D), b) the corresponding $120 \times 120$ 2D face image, c) a 3D face view obtained from (a) and (b)

(a) 2.5D (b) Y (c) Cr (d) Cb

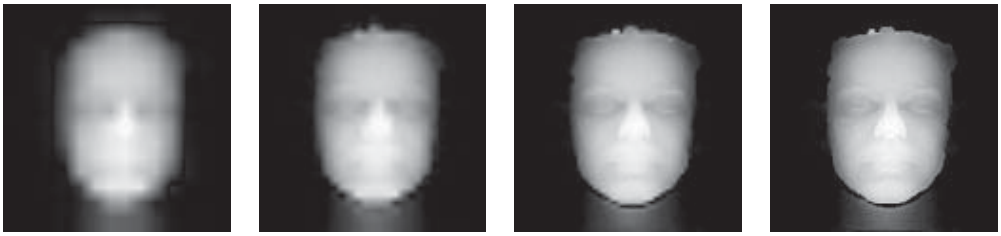
Figure 5.26: Level-3 DWT domain images: a) depth map, b-d) components of the transformed 2D face image from the lossless JPEG2000 coding pipeline.

As already stated, level-$L'$ approximate image is the one that is constructed with $(1/4^L) \times 100$ percent of the total coefficients that corresponds to the available lowest frequency $3(L-L')+1$ subbands. The level-3 encoded image with our method can give us four different quality 2D/2.5D pairs upon decoding and reconstruction. In terms of increasing quality, these are level-3, 2, 1 and 0 images reconstructed from 1.62%, 6.25%, 25% and 100% of the transmitted coefficients, respectively. The number of lowest subbands involved being $1, 4, 7$ and 10 out of the total of 10 subbands, respectively. For visual comparison, the approximation 2D images are given in Fig. 5.27 while the approximation depth maps are shown in Fig. 5.28.

For the purpose of quantitative comparison the mean results over all the FRAV3D $2D/2.5D$ pairs subjected to our method are tabulated in Table 5.3 and

(a) Level-3 (b) Level-2 (c) Level-1 (d) Level-0

Figure 5.27: 2D approximation images obtained by the decoding and reconstruction.



(a) Level-3 (b) Level-2 (c) Level-1 (d) Level-0

Figure 5.28: 2.5D approximation images after the decoding and reconstruction.

(a) Level-3 (b) Level-2 (c) Level-1 (d) Level-0

Figure 5.29: A 3D view from the visualization with the 2D/2.5D approximation pairs at different levels.

Table 5.4. For the 2D face images it can be observed that the level-3 approximate

| Approximation image | lev. 3 | lev. 2 | lev. 1. | lev. 0 |
|---|---|---|---|---|
| Bitrate (*bpp*) | 0.41 | 1.10 | 3.55 | 8.38 |
| MSE | 120.25 | 43.35 | 21.17 | 20.16 |
| PSNR (*dB*) | 27.33 | 31.76 | 34.87 | 35.09 |

Table 5.3: Results obtained for 2D face image after the extraction and reconstruction as a function of the transmitted data.

| Approximation image | lev. 3 | lev. 2 | lev. 1. | lev. 0 |
|---|---|---|---|---|
| Bits per coefficient (theoretical) | 0.14 | 0.56 | 2.25 | 9 |
| Bits per coefficient (optimized) | 0.14 | 0.44 | 1.49 | 4.98 |
| RMSE | 11.33 | 7.76 | 4.73 | 0 |
| PSNR ($dB$) | 27.05 | 30.33 | 34.63 | $\infty$ |

Table 5.4: Results obtained for the depth map after the extraction and reconstruction as a function of the transmitted data.

image is the lowest quality having a mean PSNR of 27.33 $dB$ which is not bad in the face of the fact that it is constructed from just 0.25% of the transmitted coefficients. The level-0 approximate face image has a mean PSNR of 35.09 $dB$ despite the fact that we are treating the worst case, i.e. both the 2D and 2.5D have the same dimensions. Even doubling the 2D dimensions, i.e. one depth map point corresponds to four 2D pixels, gave us a PSNRs in the order of 45 $dB$. For 2.5D approximations we are comparing the theoretical or worst case compression to that obtained by the application of our method in Table 5.4. It can be seen that for very high frequency the probability of zero is high and that is why for level-0 approximation we observed a mean bitrate of 4.98 against the expected value of 9. Since level-3 approximation has only the lowest frequency subband, the bitrate stays at 0.14 for both. We have used root mean square error (RMSE) as an error measure in length units for $2.5D$. The 3D visualization obtained from the approximation $2D/2.5D$ pairs are depicted in the form of a 3D view at a particular angle in Fig. 5.29.

### 5.4.6 Synopsis

The results have been interesting in the sense that even with a depth map of the same dimensions as the 2D face image one got a good quality visualization. Usually the sizes are not the same and one depth map coefficient corresponds to a square block of 2D face texture pixels. Even for a $2 \times 2$ block the PSNR for level-0 jumps from 35.09 dB to a maximum of 49.05 dB. The trend in our results shows that an effective visualization is possible even with a 0.1% of the transmitted coefficients, i.e. level-5. This must bode well for videoconferencing and videosurveillance applications when frames would replace the still image. Hence for a client with meager computing, memory or network resources a tiny fraction of the transmitted data should do the trick. The scalability aspect can then hierarchically take care of resourceful clients. This work had been the basis of our publication [Hayat 2009e].

## 5.5 Summary

The works presented in this chapter offer 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 uti-

lization of JPEG2000 supported DWT. Third comes 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. In addition, the approach is cost effective in terms of memory and bandwidths. It is interesting to note even the worst case scenarios gave us satisfactory results. This can be witnessed from the face example where the depth map approached the texture in terms of size.

Because of the sensitive nature of DEM it will be expedient to explore other ways to avoid an important decrease of its quality. In such situations losing details of texture may become less important than those of the DEM. Using an algorithm that would adapt the synchronization should augur well for the improvement of DEM quality. This may be specially useful for terrain visualization where there is significant difference in the sizes of DEM and texture to allow for much smaller texture blocks for embedding. In Chapter 6, we would be adapting the synchronization in embedding by limiting the latter to a subset of subbands rather than all. Another aspect of this chapter was the thrust on perceptual transparency in the form of LSB embedding. It would be worthwhile to include another important aspect of robustness to it. This is achieved in the next chapter in the form of SS embedding.

Chapter 6

# Adapting the Synchronization in Embedding

Contents



## 6.1 Introduction

*Dans le Chapitre 5, la synchronisation de la carte de profondeur et de la texture a été effectuée de telle sorte que les niveaux de décomposition du message (la carte de profondeur transformée) et du transporteur (la texture transformée) étaient les mêmes avant l'insertion. L'analyse des résultats montrent que la qualité visuelle est meilleure pour la texture que pour la carte de profondeur. Du fait de la sensibilité de la carte de profondeur, il nous a semblé intéressant d'explorer d'autres approches afin d'éviter une diminution importante de la qualité de la carte de profondeur. La différence significative des tailles de la carte de profondeur et de la texture nous autorise à utiliser de plus petits blocs de texture pour l'insertion. L'idée présentée dans ce chapitre consiste donc à augmenter la qualité de la reconstruction de la carte de profondeur dans le cas d'une transmission à bas débit. Précédemment, la*

*taille des blocs de texture était beaucoup plus élevée que la taille d'un seul coefficient de la carte de profondeur en bits. Étant donné que le nombre requis de blocs est fixe, la réduction de ces tailles permet automatiquement d'exclure certaines sous-bandes de hautes fréquences du processus d'insertion. La stratégie consiste alors à utiliser un sous-ensemble au lieu de toutes les sous-bandes de la texture pour l'insertion. En d'autres termes, la synchronisation de l'insertion peut maintenant être* ***adaptée*** *à un sous-ensemble de sous-bandes plutôt qu'à toute la texture. Le nombre de décomposition de la transformée de la carte de profondeur est alors évidemment plus faible que celui de la texture. Du fait que le flux JPEG2000 positionne en priorité les paquets des sous-bandes des basses fréquences, celles-ci sont utilisées en premier pour l'insertion. Pour répondre à la diversité des clients et afin de synchroniser les données disparates nécessaires à la visualisation 3D, nous proposons, dans ce chapitre, une stratégie synchrone adaptative et évolutive par étalement de spectre pour l'insertion des données cachés. Comme dans le chapitre précédent, nous nous appuyons sur le caractère multi-résolution du format JPEG2000 et l'insertion est utilisée afin d'unifier en un seul fichier toutes les données. La méthode proposée pour l'insertion des données cachées est aveugle, robuste et imperceptible. La robustesse est offerte par étalement de spectre et l'imperceptibilité est fournie par le caractère effacable de la méthode. En ce qui concerne la visualisation 3D, pour des raisons de robustesse, dans ce chapitre, nous remplaçons l'insertion basée sur les bits de poids faibles par une stratégie d'étalement de spectre. Cette adaptation est assurée par une analyse du point de vue afin de déterminer l'erreur admissible dans la carte de profondeur.*

In Chapter 5, the synchronized unification of the range data and texture was carried out in a manner that DWT decomposition level of both the message (transformed depth map) and the carrier (transformed texture) were the same before embedding. The analysis of the results show that the visual quality is better for the texture than for the depth map. Because of the sensitive nature of the depth map, it will be expedient to explore other ways to avoid significant decrease of its quality. The significant difference in the sizes of range data and texture allows us to use much smaller texture blocks for embedding. The idea presented in this chapter is to increase the quality of the reconstructed depth map for a low bitrate transmission. Formerly, block size of the texture have been much higher than the size of a single range coefficient in bits. Since the number of required blocks is fixed, size reduction will automatically exclude some higher subbands from the embedding process. The strategy has to be then to use a subset, rather than all the texture subbands of wavelet decomposition, for embedding. In other words, the synchronization in embedding must now be *adapted* to the subset rather than the whole of texture. The level of DWT decomposition of the depth map would then obviously be lesser than that of the texture. Since the packet order in a JPEG2000 stream is from low to high subbands, one is compelled to prefer the lowest subbands for embedding. To cater for the diverse client and unify the disparate 3D visualization data, we are proposing, in this chapter, an adaptively

synchronous scalable spread spectrum (A4S) data hiding strategy. As done in the last chapter, we are relying on the multiresolution character of the JPEG2000 format and data hiding is being employed to unify data into one file. The proposed data hiding method is blind, robust and imperceptible. Robustness is offered by the spread spectrum (SS) embedding and imperceptibility is provided by the removable nature of the method. As far as 3D visualization is concerned, for the sake of robustness, we are replacing the LSB-based embedding in this chapter by opting for a SS strategy. This adaptation is provided by a level of detail (LOD) viewpoint analysis to determine the allowable error in the depth map.

The rest of the chapter is arranged as follows. Our method is explained with details in Section 6.2 while the results which we obtained are elaborated in Section 6.3. In Section 6.4 we take the allowable limit of adaptation and apply the method to a set of 3D texture/range data pairs and analyze the robustness offered by our method. Section 6.5 sums up this chapter.

## 6.2 The Proposed A4S Data Hiding Method

In this section, we present our method for an adaptive scalable transfer and online visualization of textured 3D data. For the embedding method, like Chapter 5, we are opting for data hiding to unify the disparate visualization data. Of the data hiding techniques, probably, the LSB-based methods offer higher perceptual transparency as far as blind non-removable embedding is concerned. This is however accompanied by a marginal loss of data. Therefore, to get the maximum in terms of the perceptual transparency and robustness we are opting for a different blind embedding strategy that may not necessarily be non-removable. To this end, it is proposed over here to employ a SS data hiding strategy pioneered by Cox *et al.* [Cox 2008]. 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, highest possible texture quality is still realizable.
An overview of the method is described in Section 6.2.1. Section 6.2.2 depicts the viewpoint analysis for the adaptation of synchronization while the embedding and the decoding processes are explained in Section 6.2.3 and Section 6.2.4, respectively. The reconstruction procedure is outlined in Section 6.2.5.

### 6.2.1 Overview of the method

Suppose a $N \times N$ texture image has a depth map of $m \times m$ coefficients. In the spatial domain, let each of the coefficient corresponds to a $t \times t$ pixel block of the related texture, where $t = \frac{N}{m}$. 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 $L' \leq L$. For embedding we interrupt the

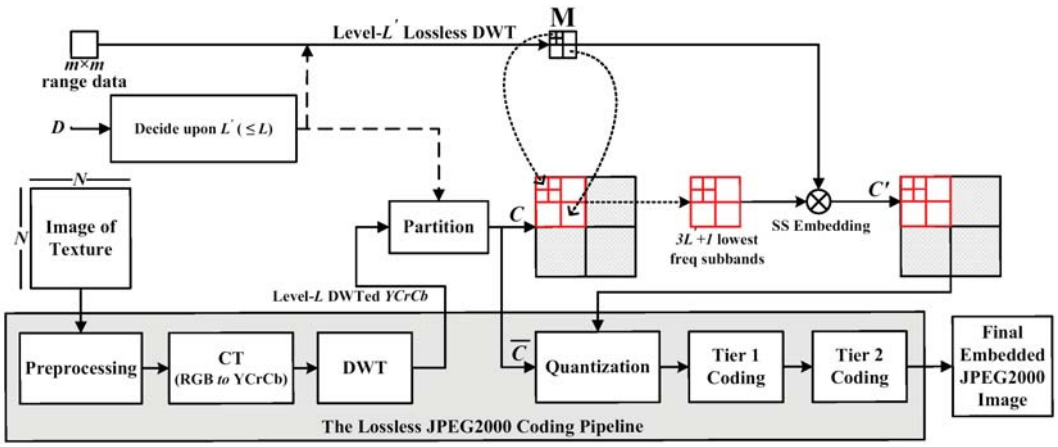



Figure 6.1: Global overview of the encoding process.

JPEG2000 coding, of the texture, after the DWT step, as illustrated in Fig. 6.1, and get one of the transformed YCrCb components as a carrier for embedding. The carrier ($C$) is partitioned into $m \times m$ equal-sized blocks, $B_{i,j}$, with size dependent on the value of $L'$. If $L' = L$ then $C$ consists of 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 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 whereby 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 is based on the value of $L'$, the block size and the involvement or otherwise of more than one YCrCb component in embedding.

### 6.2.2 Viewpoint analysis for the extent of adaptation

Our strategy is aimed at the best 3D 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 [Vepakomma 2008, Chauve 2007], even today the acquisition of the DEM, for 3D terrain visualization, is error prone and it is difficult to get a RMSE less than 1 meter. To calculate the maximum acceptable RMSE, for

an optimal 3D visualization, we rely on the distance $D$ between the viewpoint and the depth map, as illustrated in Fig. 6.2, and the visual acuity (VA) of the human visual system (HVS). 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 [Smith 1997, Luebke 2002]. In this work, for the HVS, we assume that the VA corresponds to an arc $\theta_{VA}$ of 1 minute ($1' = 1/60°$). For a distance $D$ the level of detail (LOD) could then be:

$$LOD = 2 \times D \times \tan(\theta_{VA}). \tag{6.1}$$

For example, in the case of terrain visualization, if $D = 3\ m$ then $LOD = 87{\times}10^{-4}\ m$ and if $D = 4\ km$ then $LOD = 1.164\ m$. For our application, illustrated in Fig. 6.2, if we want to see all the depth map we have a relation between $D$ and the size of the range data (width = $E$ meters):

$$E = 2 \times D \times \tan(\theta/2). \tag{6.2}$$

Usually the field of view of the HVS is $\theta = 60°$. For example if $E = 3200\ m$ then $D = 2771.28\ m$. For the 3D visualization, we should take into account the

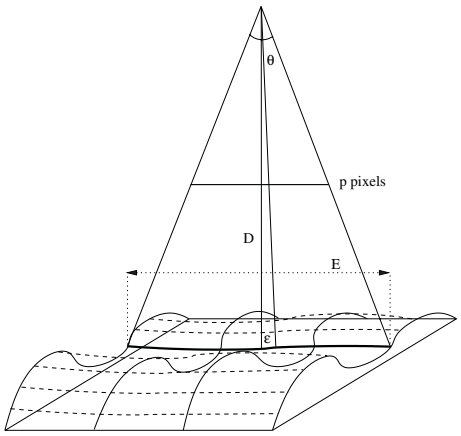



Figure 6.2: Visualization of a depth map from a viewpoint.

resolution of the screen in pixels. As illustrated in Fig. 6.2, if we have an image or a screen with $p$ pixels for each row, then the maximum LOD error ($\varepsilon$) in $p$ is:

$$\varepsilon = \frac{E/2}{\tan(\theta/2)} \times \tan(\theta/p). \tag{6.3}$$

With $\theta = 60°$, $E = 3200\ m$ and a resolution of 320 pixels (for a PDA for example), we have a $\varepsilon = 9.07\ m$. Then, in this context, for our application we can assume that a RMSE near 9 $m$ is acceptable for the DEM in case of the 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 [Puech 2001] for a particular case of a cylinder. Anyhow, the value of $D$ would help us to reach a decision about the value of $L'$.

### 6.2.3 The embedding step

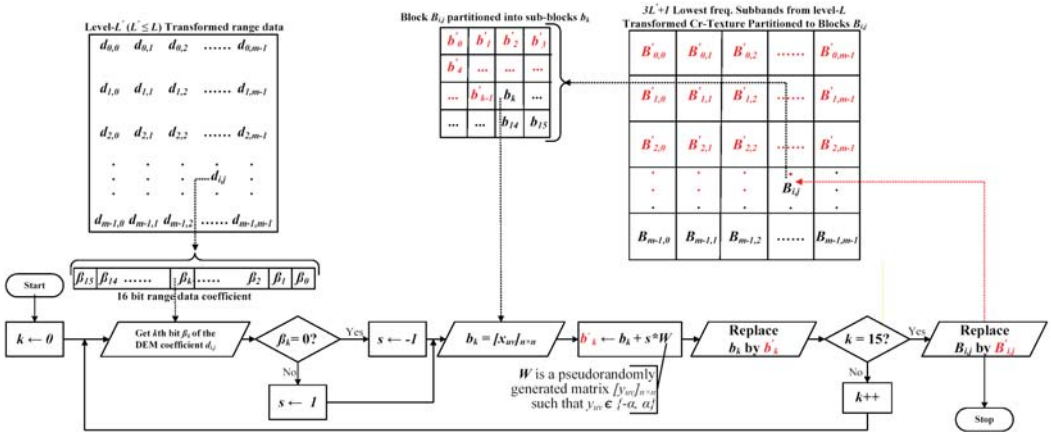



Figure 6.3: Flowchart showing the DWT domain SS embedding of the range data coefficient $d_{i,j}$ in the corresponding block partition $B_{i,j}$ of the lowest $3L'+1$ subbands of a component of texture.

The process of embedding of a given DWT-domain range data coefficient in 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 $\omega$, where $\omega$ 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 the algorithm 2 proposes a solution to choose only one or three components. The choice whether to embed in Y or Cr/Cb plane depends on the fact that Y plane

**Algorithm 2** A sub-routine to choose component(s) for embedding.

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

embedding would distort the encoded image while the 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 (size $= t^2/2^{2(L-L')}$) is elaborated by the flowchart given in Fig. 6.3. 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 sub-blocks ($b_k$) as are the number of bits used to represent a single transformed coefficient. Each of $b_k$ from $B_{i,j}$ would carry the $k^{th}$ bit $\beta_k$ of the transformed range altitude $d_{i,j}$. Embedding depends on a key generating a pseudo-random matrix $W$ with entries from the set $\{\alpha, -\alpha\}$. The matrix $W$ has the same size as of $b_k$, i.e. $W = [y_{uv}]_{n\times n}$, where $y_{uv} \in \{\alpha, -\alpha\}$. The scalar, $\alpha$, is referred to as the strength of embedding. If the bit to embed, $\beta_k$, is '1', then matrix $W$ is added to the matrix $b_k$, otherwise $W$ is subtracted from $b_k$. The result is a new matrix $b'_k$ which replaces $b_k$ as a sub-block in the embedded block, $B'_{i,j}$, of the marked carrier $C'$. Two factors are important

**Algorithm 3** Embedding of the range data coefficient $d_{i,j}$ in the corresponding block partition $B_{i,j}$.

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 sub-blocks, $b_0, b_1, \ldots b_{15}$
4: **for** $k \leftarrow 0$ *to* 15 **do**
5: **read** the $k$th bit $\beta_k$ of the DEM coefficient $d_{i,j}$
6: **if** $\beta_k = 0$ **then**
7: **set** $b'_k \leftarrow b_k - W$ /*$W$ is a one-time pseudorandomly constructed matrix*/
8: **else**
9: **set** $b'_k \leftarrow b_k + W$
10: **replace** $b_k$ by $b'_k$ in the block $B_{i,j}$ which will ultimately change to $B'_{i,j}$
11: **replace** $B_{i,j}$ by $B'_{i,j}$ in the cover
12: **end**

over here. First is the DWT level ($L'$) of transformed range data before embedding which is a trade off 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, $(L - L')$, higher will be the quality and vice versa. Second is the value of $\alpha$ since a larger $\alpha$ 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 highest possible quality.

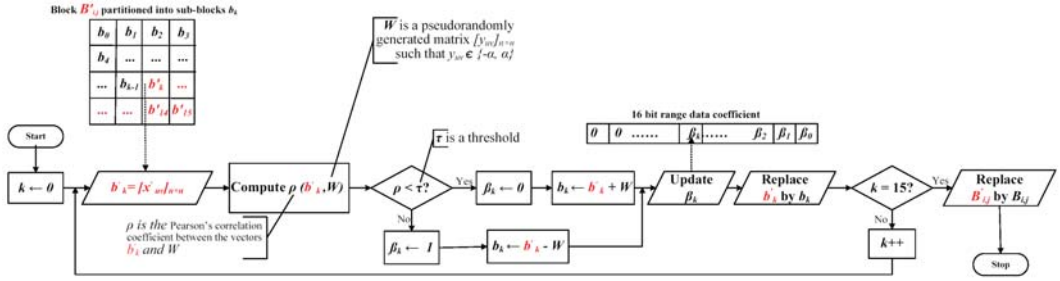



Figure 6.4: Flowchart showing the recovery process of the range data coefficient $d_{i,j}$ and the corresponding block $B_{i,j}$ from the SS watermarked texture block partition $B'_{i,j}$ of the lowest $3L'+1$ subbands of a watermarked texture component.

### 6.2.4 The decoding step

The above 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 above process. Just before the inverse DWT stage of the JPEG2000 decoder the range data can be extracted from $C'$, using the above mentioned partitioning scheme, i.e. $B'_{i,j}$ blocks and their $b'_k$ sub-blocks. Fig. 6.4 shows the flowchart for the recovery of a range altitude $d_{i,j}$ from sub-blocks $b'_k$ of $B'_{i,j}$ and eventual reconstruction of $b_{i,j}$ and ultimately $B_{i,j}$. A given $k^{th}$ sub-block $b'_k$ and also the matrix $W$ can be treated as a column/row vector and the Pearson's correlation coefficient, $\rho$, can be computed thereof. If $\rho$ is less than certain threshold, $\tau$, then the embedded bit $\beta$ 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 $\beta$ is '0' then add $W$ to $b'_k$, otherwise subtract $W$ from $b'_k$.

### 6.2.5 The reconstruction step

Now comes the reconstruction phase wherein by the application of 0-*padding* one can have $L+1$ and $L'+1$ different approximation images of texture and its range data, respectively. And this is where one achieve 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 $3D$ visualization even with fewer than original resolution layers as a result of partial or delayed data transfer. The method thus enables 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 0's for the higher bands. The idea is to have a $3D$ visualization utilizing lower frequency subbands at level $L''$, say, where $L'' \leq L$. For the rest of $L - L''$ parts one can always pad a 0 for each of their coefficient. The inverse DWT of the 0-stuffed

transform components will yield what is known as image of approximation of level $L''$. Before this, as depicted by algorithm 4, data related to the third dimension, i.e. range data, must be extracted whose size depends on both $L''$ and $L'$. Thus if $L' \leq L''$ 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 0's for all coefficients of higher $L' - L''$ subbands of transformed range map before inverse $DWT$ that would result in a level $L''$-approximate range data of an inferior quality.

**Algorithm 4** 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 which may correspond to $L''$ or a larger level $L'' + \Delta$, where $\Delta \geq 0$ depends on the extent of adaptation
4: **for** both the texture and the 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**

## 6.3 Experimental Results

We have applied our method to examples from two areas, namely the terrain and facial visualizations, with the results are being presented and analyzed in Section 6.3.1 and 6.3.2, respectively.

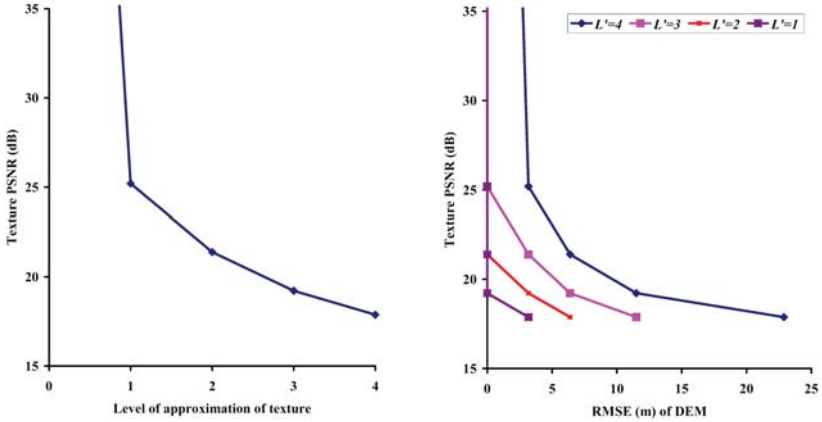
### 6.3.1 Terrain visualization

A good and elaborate application of wavelets to real-time data for terrain visualization can be found in the work of Kim and Ra [Kim 2004]. 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 the lack of integration with the texture. Earlier efforts, found in the literature, in this context are already discussed in Section 3.4.1.3. The problem with these approaches is, however, the lack of synchronization. We have chosen the example texture/DEM pair given in Fig. 6.5 to try various possible adaptations in embedding. The DEM (Fig. 6.5.a) is shown in the form of a $32 \times 32$ gray scale image wherein the whiteness determine the height of the altitude. The corresponding texture (Fig. 6.5.b) has a size of $3072 \times 3072$ pixels implying one DEM coefficient per $96 \times 96$ texture block, i.e. $t = 96$. For the purpose of comparison a $256 \times 256$ pixel portion, at (1000, 1500) coordinates, is magnified as shown in Fig. 6.5.c. Fig. 6.5.d illustrates a 3D view ob-

tained 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 $1, 4, 7, 10$ lowest frequency or all of the 13 subbands.

(a) (b) (c) (d)

Figure 6.5: Example images: a) $32 \times 32$ DEM, b) $3072 \times 3072$ pixel texture, c) a $256 \times 256$ pixel magnified detail of (b) at (1000, 1500) coordinates, d) a corresponding 3D view.




(a) (b)

Figure 6.6: Texture quality as a function of: a) level of approximation of texture, b) RMSE of DEM in meters.

For fully synchronous embedding, all the 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, $B_{i,j}$, had then a size of $96 \times 96$ for the embedding of one 16 bit transformed DEM coefficient which means a $24 \times 24$ sub-block ($b_k$) per DEM bit. Since $b_k$ is large enough, the needed strength/amplitude ($\alpha$) of embedding is smaller and the quality of luminance/chrominance component will not be deteriorated 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* as against 14.55 *bpp* for any adaptation. The texture bitrate is thus independent of the extent of adaptation in partially synchronized cases. This is probably due to the exclusion of three largest subbands from embedding; these bands, incidently, have the highest probability of zero coefficients.

(a) (b) (c) (d)

Figure 6.7: Coded texture after embedding in: a) Y plane ($\alpha = 64$) with PSNR= 13.40 dB, b) its $256 \times 256$ pixel magnified detail, c) Cr plane ($\alpha = 16$) with PSNR= 31.47 dB, and d) its $256 \times 256$ pixel magnified detail.

Upon the reintroduction of the marked Y or Cr/Cb to the JPEG200 pipeline we get our watermarked texture image in JPEG2000 format. From this image one can have 5 different approximation images for both the DEM and the texture. A level-$l$ ($\leq L$) approximate image is the one that is constructed with $(1/4^l) \times 100$ percent of the total coefficients that corresponds to the available lower $3l + 1$ subbands. For example, level-0 approximate image is constructed from all the coefficients and level-2 approximate image is constructed from 6.12% of the count of the initial coefficients. Since the embedded data is 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. Fig. 6.6.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 are the main causes.

The sensitive nature of DEM compels us to avoid too much loss in its quality. For improved DEM quality one had to adjust the synchronization and rather than persisting with $L' = 4$ we went for $L' < 4$, i.e. 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

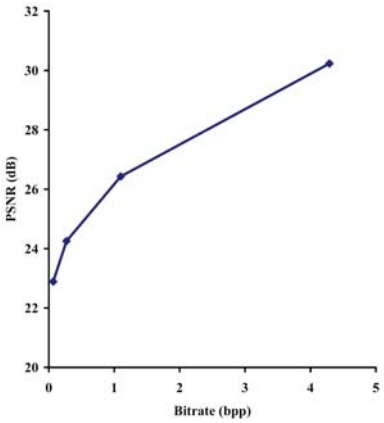



Figure 6.8: Variation in texture quality as a function of its bitrate for a DEM error of 3.18 $m$.

$3L' + 1$ 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 $\alpha$ and an eventual degradation in the quality of the coded texture. As described in Section 6.2.4, 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 the fact that how much distortion in the carrier is reversible, i.e. bounds of $\alpha$. For our example, the lowest quality for the coded texture - 13.40 dB with $\alpha = 64$ for Y carrier (Fig. 6.7.a-b) and 31.47 dB with $\alpha = 16$ for Cr carrier (Fig. 6.7.c-d) - was observed at $L' = 1$. The reason is that all the information had to be embedded in just 4 lowest energy and smallest subbands implying a size of $3 \times 3$ for $b_k$ with $\alpha = 64$ for Y carrier and $\alpha = 16$ for 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. Fig. 6.6.b illustrates the trend in texture quality as a function of the 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 Section 6.2.2, 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 $L'$, higher is the degree of adaptation in synchronization and hence higher is the resultant DEM quality.

With our approach, by using the example given in Fig. 6.5, the RMSE of the five possible DEM approximations, i.e. level $0, 1, 2, 3$ and 4 were found to be $0\ m, 3.18\ m, 6.39\ m, 11.5\ m$ and $22.88\ m$, respectively. For full synchronization, as presented in [Hayat 2009a], the worst DEM quality, $22.88\ m$, is resulted when one goes for a 3D visualization from level-4 approximate texture as the corresponding DEM will also be 4-approximate. But even with one step adjustment this quality is twice improved and for $L' = 3$, both the 3 and 4 approximate texture images have 3 approximate DEM with RMSE $= 11.5\ m$. Go a step further and maximum DEM

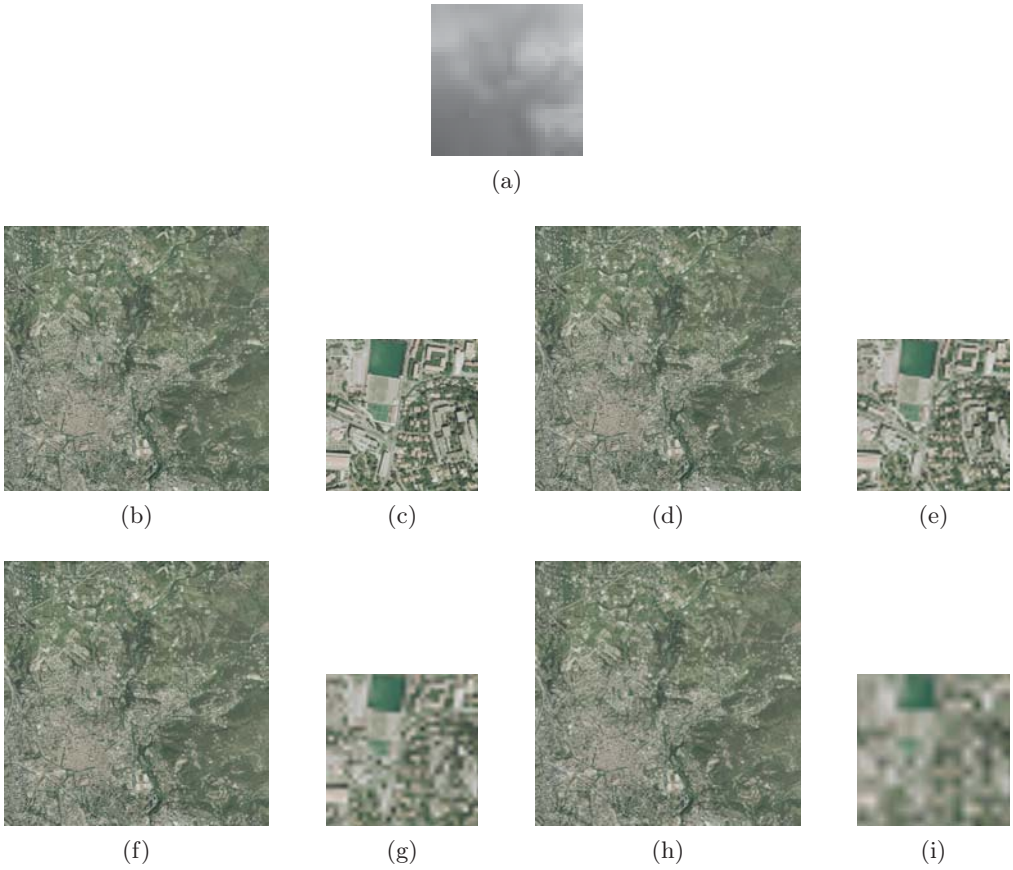
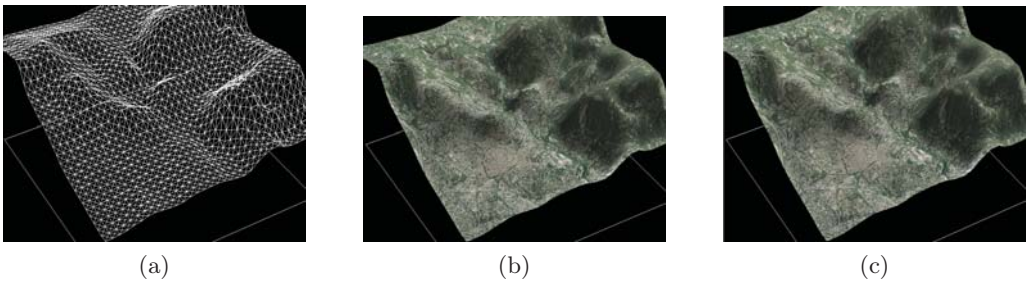



Figure 6.9: Approximation images corresponding to 3.18 $m$ DEM error when level-$L'$ transformed DEM is embedded in lowest $3L'+1$ subbands of level-4 transformed Cr texture: a) level-1 approximate DEM having 3.18 $m$ error, b) level-1 approximate texture (4.29 bpp) with $L' = 4$, c) its $256 \times 256$ pixel magnified detail, d) level-2 approximate texture (1.1 bpp) with $L' = 3$, e) its $256 \times 256$ pixel magnified detail, f) level-3 approximate texture (0.27 bpp) with $L' = 2$, g) its $256 \times 256$ pixel magnified detail, h) level-4 approximate texture (0.066 bpp) with $L' = 1$ and i) its $256 \times 256$ pixel magnified detail.

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 for all the approximation images at levels $\geq L'$. Fig. 6.8 shows the trend in texture quality for a given DEM error (3.18 $m$) as a function of bitrate. Images corresponding to a DEM error of 3.18 $m$ are shown in Fig. 6.9 and the resultant 3D visualizations are illustrated in Fig. 6.10. It can be seen that with even a tiny fraction of the total coefficients (as low as 0.40%, i.e. Fig. 6.9h-i and Fig. 6.10.e), a fairly commendable visualization can be realized.

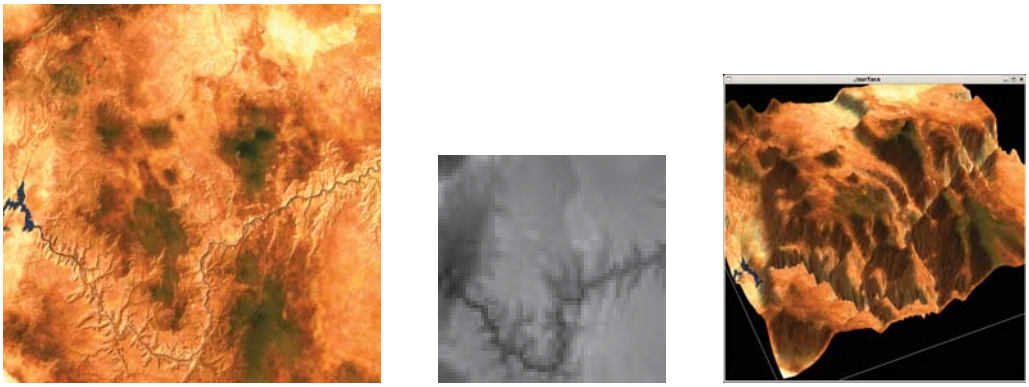
The maximum error in RMSE of the DEM tolerable by an observer is a function of the distance of the observer, i.e. the viewpoint $D$. This threshold is mainly dependent on the human visual system (HVS) and for this reason, the analysis



(a) (b) (c)

(d) (e)

Figure 6.10: 3D visualization corresponding to Fig. 6.9: a) level-1 approximate DEM having 3.18 $m$ error, b) level-1 approximate texture (4.29 bpp) with $L' = 4$, c) level-2 approximate texture (1.1 bpp) with $L' = 3$, d) level-3 approximate texture (0.27 bpp) with $L' = 2$, e) level-4 approximate texture (0.066 bpp) with $L' = 1$.



(a) Texture (b) DEM (c) A 3D view

Figure 6.11: The Grand Canyon terrain example.

given in Section 6.2.2 is extremely useful to adapt the synchronization.
We had also applied the method to a completely different terrain example of Grand Canyon[1]. We have trimmed the original example to the one in Fig. 6.11, with a $2048 \times 2048$ pixel texture and a $64 \times 64$ coefficients DEM. Due to it larger size we have increased the decomposition level by one further degree ($L = 5$) with $L' = 4$ and $L' = 5$ for the desynchronization and synchronization cases respectively. The

[1] http://www.cc.gatech.edu/projects/large_models/gcanyon.html

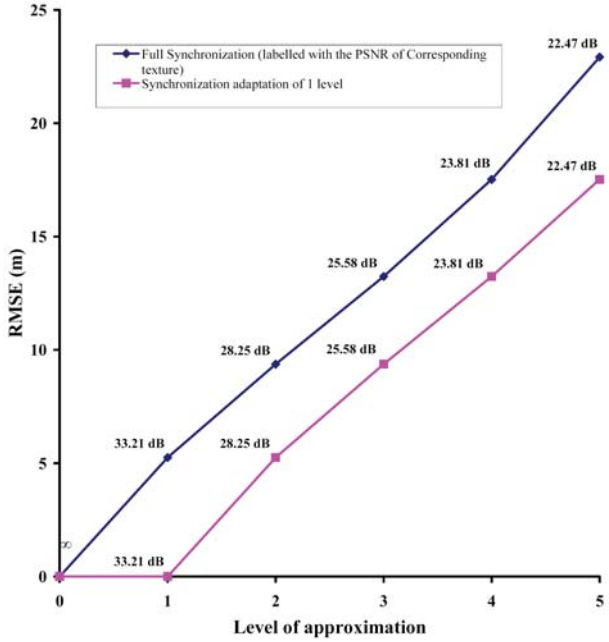
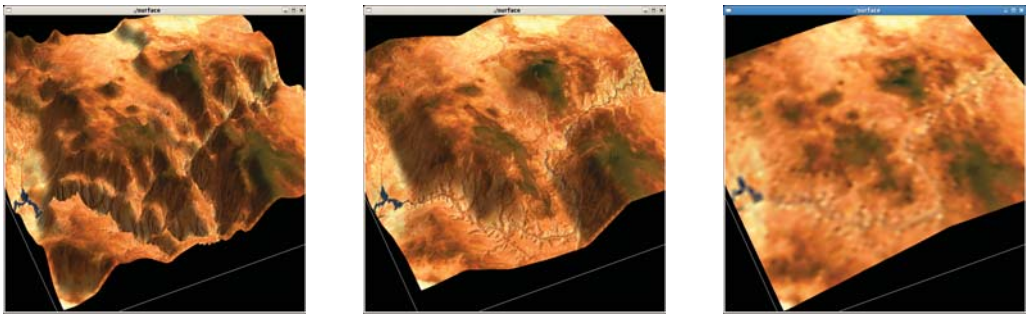


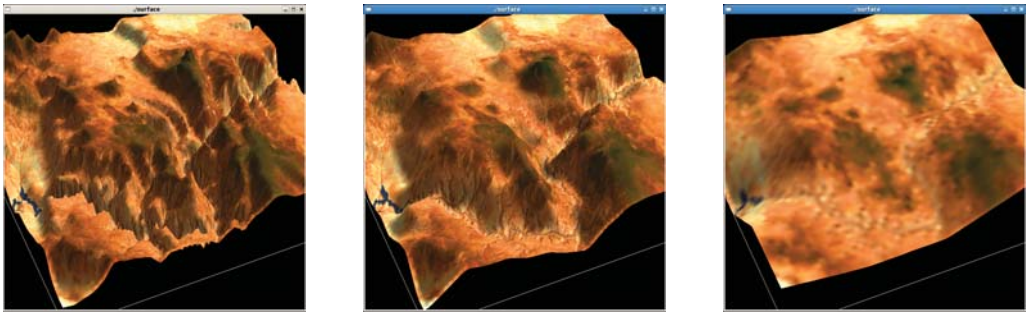

Figure 6.12: Variation in RMSE of the DEM, from the Grand Canyon example, as a function of its level of approximation with each point labeled with the PSNR of the corresponding texture

quality difference in the DEM at various approximation levels can be seen in Fig. 6.12 where RMSE of DEM is plotted against the level of its approximation. Each data point is labeled with the PSNR of corresponding texture. The data points are again labeled with their corresponding texture quality. It is obvious that even one level adaptation, in synchronization, markedly improves the DEM quality for the same texture quality.



(a) Level-1 (b) Level-3 (c) Level-5

Figure 6.13: 3D visualization corresponding to various approximations of the Grand Canyon example realized through the overlaying of level-$l$ approximate texture on the extracted level-$l$ approximate DEM.



(a) Level-1 (b) Level-3 (c) Level-5

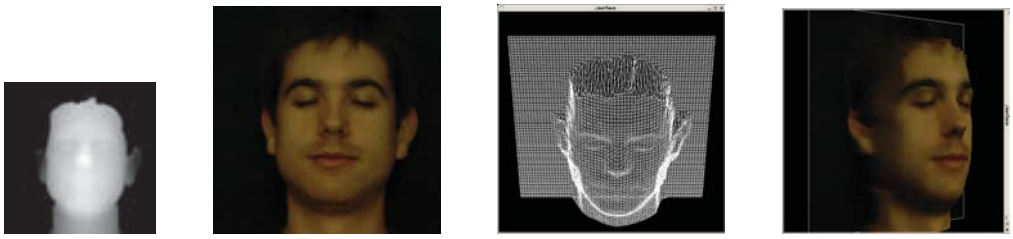
Figure 6.14: 3D visualization corresponding to various approximations of the Grand Canyon example realized through the overlaying of level-$l$ approximate texture on the extracted level-$(l-1)$ approximate DEM.

A comparison of 3D visualizations illustrated in Fig. 6.13 and Fig. 6.14 suggests that the latter is always better by one level (four times, roughly) in terms of DEM quality. 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 level-$l-1$ approximate depth map as compared to the relatively inferior level-$l$ approximate depth map for the fully synchronized case.

### 6.3.2 Application to 3D face visualization

For the case of the face visualization data we applied our method to a set of examples from FRAV3D[2] database, each consisting of a $512 \times 512$ face texture and a corresponding $64 \times 64$ depth map of 8-bit coefficients [?]. An example is given in Fig. 6.15. Thus one depth map coefficient corresponds to a $8 \times 8$ 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 was subjected to level-$L'$ lossless DWT before embedding in the three DWT-domain components from the JPEG2000 coder. To ensure the accuracy we expand the word-size for each of the transformed depth map coefficient by one additional bit and represent it in 9 bits which are then equally distributed amongst 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 5 coefficients of Y, Cr or Cb plane in the DWT domain. For the set of face examples, the maximum of the lowest value of $\alpha$ needed for the 100% recovery of the embedded data was found to be 26 for the synchronous case and 45 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

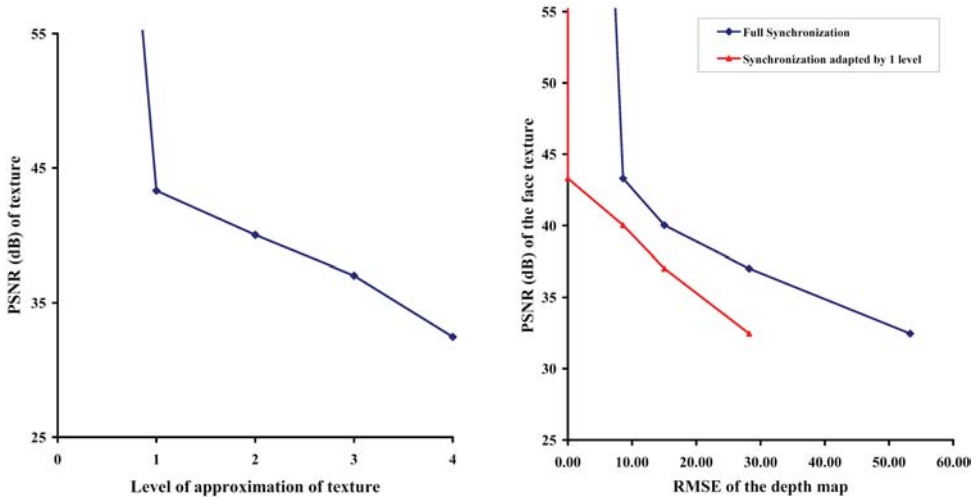
[2] http://www.frav.es/databases/FRAV3d/



(a) (b) (c) (d)

Figure 6.15: Example face visualization: a) $64 \times 64$ depth map, b) $512 \times 512$ pixel face texture, c) the corresponding 3D mesh, d) a 3D view.

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. Fig. 6.16.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 arround 30 $dB$. The advantage of adaptation is evident from the fact that, as shown in Fig. 6.16.b, for the same texture quality, even one level adaptation offers far better depth map quality as compared to full sychronization. The evident edge of the adaptive strategy in




(a) (b)

Figure 6.16: Variation in the texture quality as a function of: a) its approximation level, b) RMSE of the extracted depth map.

depth map quality is conspicuous by the comparison of 3D visualizations illustrated in Fig. 6.17 and Fig. 6.18. 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 level-$l-1$ approximate depth map

as compared to the relatively inferior level-$l$ approximate depth map for the fully synchronized case.

(a) $l = 1$ (b) $l = 2$ (c) $l = 3$

Figure 6.17: 3D visualization corresponding to Fig. 6.15 realized through the overlaying of level-$l$ approximate texture on the extracted level-$l$ approximate depth map.

(a) $l = 1$ (b) $l = 2$ (c) $l = 3$

Figure 6.18: 3D visualization corresponding to Fig. 6.15 realized through the overlaying of level-$l$ approximate texture on the extracted level-$(l-1)$ approximate depth map.

## 6.4 Robustness analysis

For the sake of brevity we will use the nomenclature described in Table 6.1 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 examples corresponding to the region of *Bouches du Rhône* in France provided by the IGN[3] France. All the examples were

[3] www.ign.fr

| 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 6.1: Nomenclature.

composed of a $1024 \times 1024$ pixel texture and a corresponding DEM of $32 \times 32$ 16 bit altitudes. All the textures were subjected to level-4 ($L = 4$) decomposition in the JPEG2000 codec. The DEM were DWTed at level $L' = 4$ for synY and synCr, while for desynY and desynCr we are opting for one level adaptation by choosing $L' = 3$. For each pair we determined minimum of $\alpha$ needed for 100% recovery of the embedded coefficients. For all such $\alpha$'s we determined the minimum ($\alpha_{min}$) and maximum ($\alpha_{max}$) in the set images for each case, as given in Table 6.2. It can be observed from the PSNR values given in the table that the Y-plane embedding and larger $\alpha$ 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 6.2 for each of the four embedding cases. The watermarked JPEG2000 coded textures were then subjected to robustness attacks.

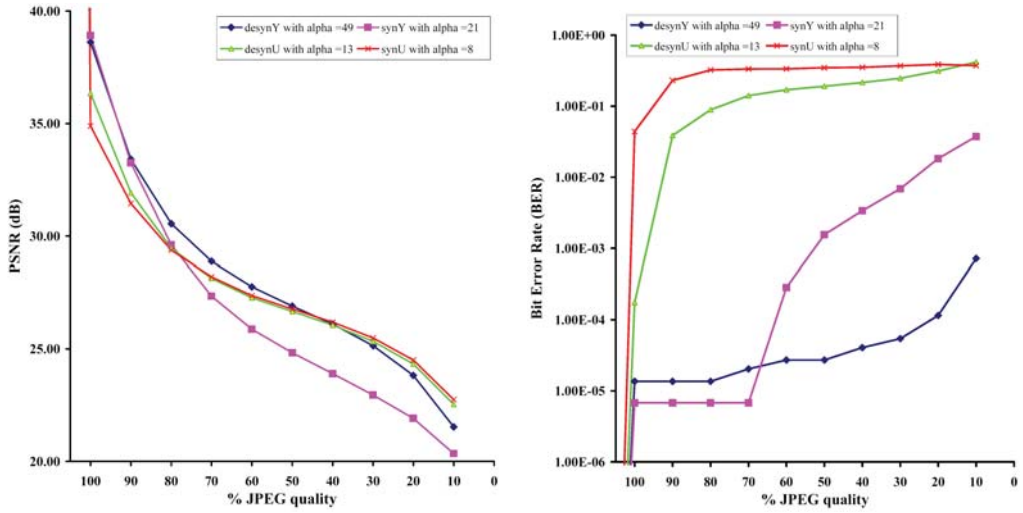
| Case | synY | synCr | desynY | desynCr |
|---|---|---|---|---|
| $\alpha_{min}$ | 12 | 32 | 4 | 10 |
| $\alpha_{max}$ | 21 | 49 | 8 | 13 |
| **Mean PSNR** (coded texture) | 18.25 *dB* | 12.90 *dB* | 32.94 *dB* | 30.23 *dB* |

Table 6.2: Extreme values of $\alpha$ for various embedding scenarios.

### 6.4.1 Resistance to JPEG compression

To show the intensity of the attack by JPEG compression we have plotted the average PSNR of the distorted texture against the quality factor of compression, as shown in Fig. 6.19.a. It can be observed that for all the 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. 6.19.b. The reason 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 the

most sensitive to the 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, the Y-plane embedding offers high robustness , with desynY being the most robust.




(a) (b)

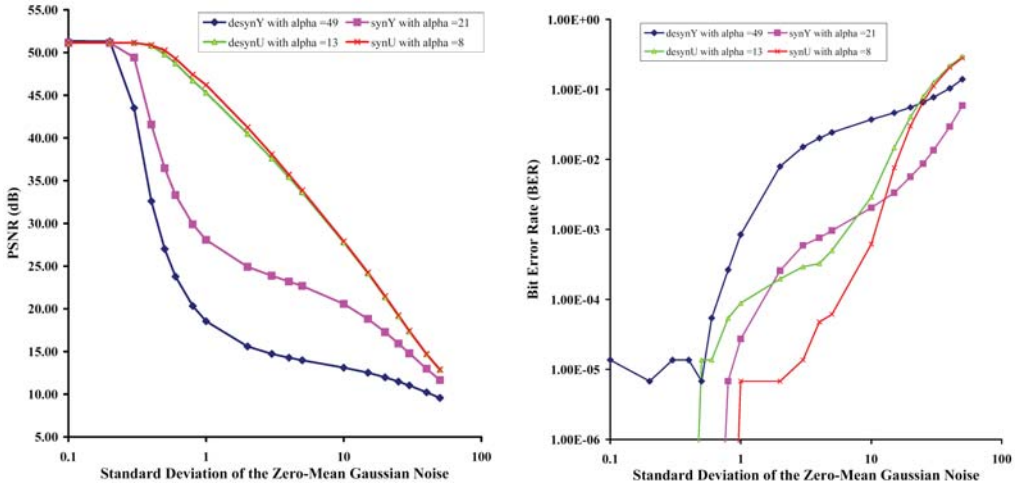
Figure 6.19: Robustness to JPEG compression: a) Quality of the watermarked texture after various JPEG quality factors, b) Bit error rate (BER) of the extracted range data after the JPEG compression of the watermarked texture.

### 6.4.2 Robustness against Gaussian noise addition

The amount of distortion as a result of the Gaussian noise can be judged from Fig. 6.20.a wherein the embedding involving Cr/Cb 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 ($\sigma$'s) in the range $0.1 - 50$. Fig. 6.20.b shows the graph whereby the mean BER is plotted as a function of $\sigma$, with both the axes drawn on a logarithmic scale. In general all the cases are robust till about $\sigma = 0.8$ which is quite a large value. Beyond this value the robustness varies from case to case with the fully synchronous cases being the most robust.

### 6.4.3 The prospects of cropping

The robustness demonstrated above 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:




(a) (b)

Figure 6.20: Robustness to a zero mean Gaussian noise at various standard deviations ($\sigma$): a) quality of the watermarked texture after the introduction of the Gaussian noise, b) bit error rate (BER) of the extracted range data after the addition of the Gaussian noise to the watermarked texture.

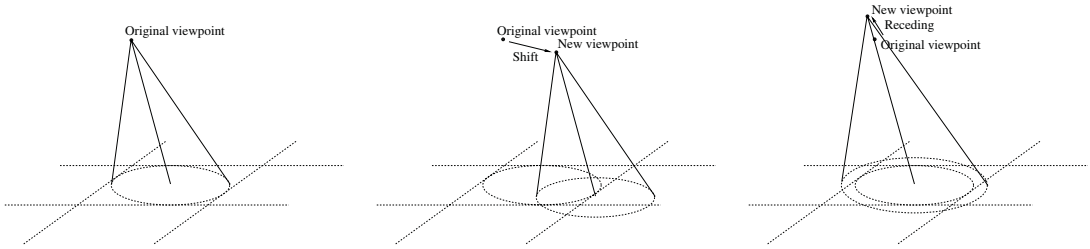
1. The cropped patch has dimensions, that are multiple of $PD$ where:

$$PD = \sqrt{1 + 3\sum_{i=0}^{L-1} 4^i} \times t/2^{(L-L')} \tag{6.4}$$

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

Calculating of $PD$ is necessary since the embedding is done in the level-$L$ DWT domain, 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 $1 + 3\sum_{i=0}^{L-1} 4^i$ 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. Recall the equations in the set 2.29 for a $1D$ (pixel row or pixel column) input signal $s_0, s_1, s_2, ..., s_{n-1}$. Both of the equations suggest that approximations are inevitable at boundaries, e.g. for computing the coefficient $L_0$ one would definitely need $H_{-1}$ which would in turn need $S_{-1}$ and $S_{-2}$, both of which do not exists and are normally approximated by $S_0$ and $S_1$, respectively. Same is the case with the upper boundary. In 2D the upper and lower boundaries of a given subband has 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 initial if the transform is supposed to be reversible. And it is here that our problem starts as 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 a critical data. The possible solutions are then either not to transform the DEM and embed in the lowest frequency DWT domain texture subband or transform at low level and crop a larger patch than absolutely needed. Both these scenarios offer limited advantage as the former would require high disparity in the sizes of texture and DEM which is normally not the case and the latter would crop multiple times more than needed.
All of the above facts notwithstanding, we have a silver lining in the form of boundary patches since they will have errors located in the two borders they do not share with the texture images from which they are cropped. To elaborate further let us take a visualization scenario where a number of texture images are to be tiled together for the visualization of 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 the 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 6.21 graphically illustrates different focus situations with the circle representing the focus and the rectangle with broken borders represent the minimum area that must be rendered for visualization. In addition to the above, cropping my be specially useful even with limited DEM coefficients in the scenario like the one explained by Lafarge *et al* [Lafarge 2008].




(a) Focus center (b) Shift focus (c) Viewer recedes

Figure 6.21: Changing of field of view.

Let us suppose the Grand Canyon example is to be rendered at tile bottom-right in the situation given in Fig. 6.21.a. Let the needed upper left corner corresponds to an area less than $256 \times 256$ pixels. Let the texture has been embedded synchronously at $L = 2$ which would mean, in the light of the discussion in the beginning of this section, that after extraction from a $256 \times 256$ pixel texture patch and then inverse DWT, the DEM patch would be accurate but for the last three rows and columns of

the coefficients. Fig. 6.22 compares the 3D visualization obtained by decoding and extracting the cropped patch by our method with that obtained from the original DEM corresponding to the patch.

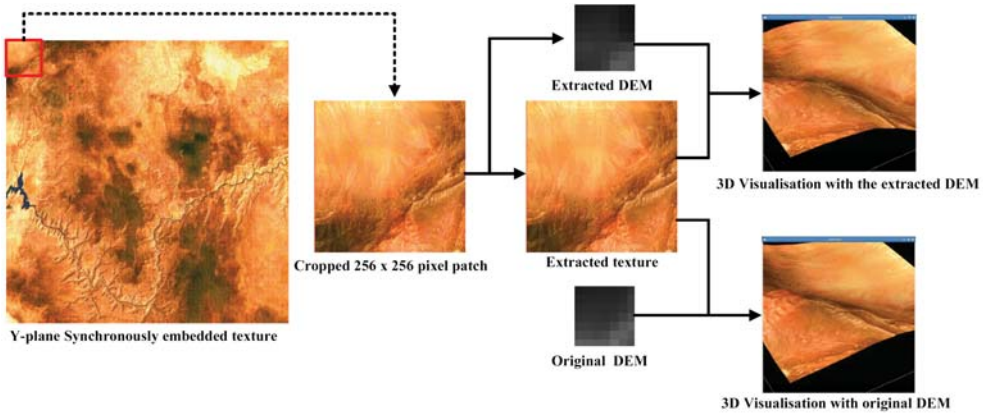



Figure 6.22: 3D visualization from a embedded patch cropped from the upper left corner of a Y plane watermarked texture.

## 6.5 Summary

In this chapter we have presented an efficient adaptive method to hide 3D data in a texture file in order to have a 3D visualization. The proposed method has the peculiarity of being robust and imperceptible, simultaneously [Hayat 2009b]. High strength of embedding may some how go against the security but the latter has never been the dominant goal. With the adaptation in synchronization, higher quality of the depth map was ensured but the extent of adaptation is strictly dependent on the embedding factor. The more the range data size approaches that of the texture, one is compelled to increase the strength of embedding. If the trend goes, however, the prospect of using SS embedding may gradually diminish before involving more than one YCrCb planes in embedding. At the extreme if SS embedding is not a viable option one can employ the LSB embedding while retaining some adaptation, as has been reported in [Hayat 2008c]. 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, specail wavelet transforms in the JPEG2000 codec by engineering some plugin. Ideally, these wavelets must offer lowest quality 'gap' between the various resolution levels. The robustness studies seem to be really interesting, especially in the case of cropping wherein an effort was made to realize a 3D visualization with a small patch is cropped from the encoded image.

Chapter 7

# Seamless Tessellation via DWT Domain Smoothing

Contents



## 7.1 Introduction


*Dans les Chapitres 5 et 6 nous avons développé des méthodes permettant la synchronisation des données pour la visualisation 3D adaptée à distance. Nous prenons en compte, dans ce chapitre, le fait d'avoir une vue plus large s'appuyant sur plusieurs dalles en se concentrant sur la jonction de dalles voisines codées séparement avec les méthodes que nous proposons. Dans ce cas, des dalles voisines risquent d'être codées et transmises avec des niveaux de résolution différents en fonction de leur distance avec le point de vue. Dans cette perspective, une nouvelle stratégie de lissage, basée sur le contexte et dans le domaine des ondelettes, est proposée afin d'obtenir des raccords transparents au niveau de la texture et de la carte de profondeur. Pour notre scenario, nous avons considéré trois niveaux de résolution différents, fonction de la distance, du débit et des ressources matérielles. Le niveau de détail est dépendant des caractéristiques multiresolution des ondelettes issues du codeur JPEG2000. Les jonctions sans raccords visibles sont réalisées par lissage orienté dans le domaine des ondelettes pendant le décodage. Trois fonctions de jonction sont présentées afin de créer des masques de la taille des sous-bandes pour effectuer le lissage. Les résultats expérimentaux obtenus semblent intéressants dans le sens où notre approche permet d'éliminer des artefacts sans affecter le champ de vue.*


In Chapter 5 and Chapter 6 we developed methods for the synchronized unification of the data for scalable 3D visualization. We take into account, in this chapter, the fact of having a broader tile-based view on focusing on the junctions of neighboring tiles which are separately coded with our proposed methods. In this

case, neighboring tiles may be encoded and transmitted with different resolution levels according to their distance from the viewpoint. With this in view, a novel context-based DWT domain smoothing strategy is being proposed for seamless tessellation of texture as well as the depth map. In the use case, we have considered three different levels of resolutions, depending on the viewpoint distance, the time of rendering and hardware resources. The level of detail (LOD) is dependent on the multiresolution characteristic of wavelets from the now widely accepted JPEG2000 codec. Seamless tiling are realized by the wavelet domain directional smoothing during the decoding. Three different joining functions are presented in order to create subband-sized masks for each of tile subbands selected for smoothing. The results, obtained for a practical example, have been interesting in the sense that it eliminated the popping artifacts to a considerable extent without affecting the field of view at a given instant.

The rest of this chapter is arranged as follows. Section 7.2 gives a brief background over view followed by the explanation of the method in terms of the scenario and the smoothing functions in Section 7.3. We applied our strategy to a practical worst case example and the results are shown in Section 7.4 which is followed by the concluding remarks in Section 7.5.

## 7.2 Background

The technology is advancing by leaps and bounds, but it is still away too far from catching up with the ever increasing rendering throughput of terrain visualization. The problem is exacerbated when the data is to be transferred over some network, especially in a client/sever environment. With it comes another dimension in the shape of the client diversity in terms of memory, processing and bandwidth resources. For example, today's geo-browsers, like Google Earth or NASA's World Wind, do not take into account individual client characteristics and what one observes are lags and freezes for clients with low resources. All these factors necessitate some scalable data processing/structuring and the obvious choice is to create multiple levels of detail (LOD) by representing the shape at different levels of approximations. This is essential for the terrain tessellation adjustment as a function of the view parameters [Losasso 2004]. For LOD one can rely on the multiresolution nature of the discrete wavelet transform (DWT). It is better to employ some standard state-of-the-art DWT, like the now widely accepted JPEG2000 standard[1].

As far as the rendering of large terrains is concerned, tile-based approaches have always been a preference. A snapshot of a Google Earth example is shown in Fig. 7.1 where the observer is changing his view to the right: it can be seen that the resolution is not uniform. Usually there are two reasons for this heterogeneity. First is the download latency and second is the non-availability of proper resolution due

[1]The ISO/IEC 15444-1 standard

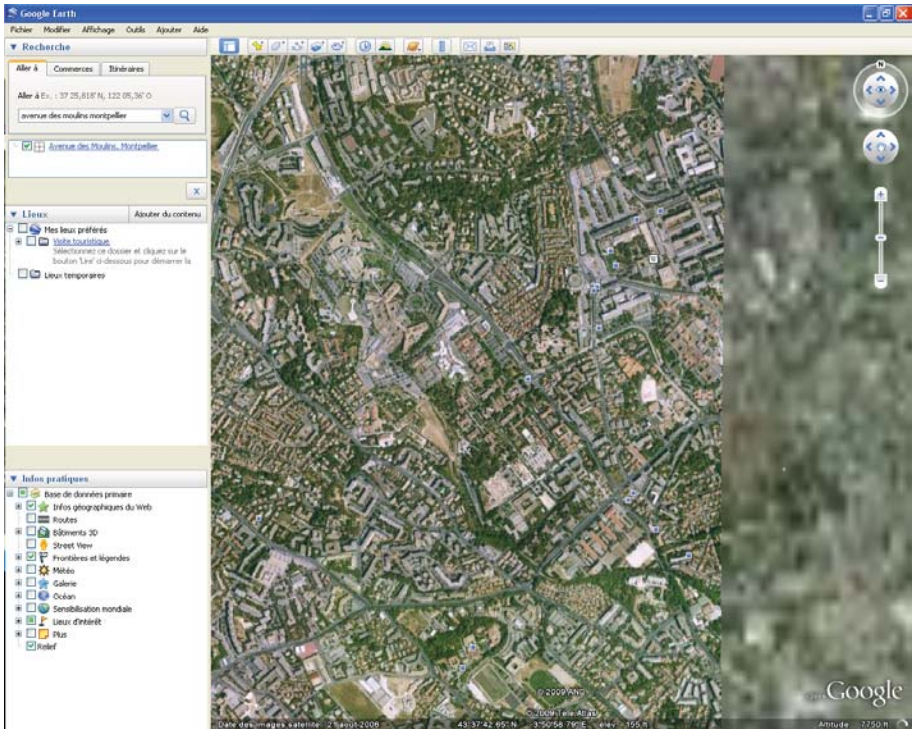



Figure 7.1: A Google Earth example.

to a variety of reasons like area security, data inaccessibility and the like. Anyhow, we have a gap between the resolutions. Our strategy is to render a texture/DEM tessellation from JPEG2000-coded tiles wherein only visible tiles are rendered and the focused ones, out of these, have the highest quality depending on the viewer's distance. As the viewer moves in, the tiles corresponding to the new view must be rendered at and gradually refined in quality as the focus is coming near to it. This refinement is additive as more and more subbands are added to it, at runtime. The main challenge to this strategy is that the tile must be seamlessly stitched for both the texture and the DEM and the boundaries must be diluted enough to prevent popping artifacts. This phenomenon is observable even in the case of modern geo-browsers. While there have been many works on the seamless rendering of DEM, the efforts focusing on the textures are far less. In this paper, although we are more inclined to the texture, the DEM has also been treated as its integral part. In fact this work is a continuation of our earlier efforts about the synchronous unification of the DEM/texture pair through data hiding [Hayat 2008b].

Our strategy is aimed at context-based smoothing of the tile boundaries. The smoothing is carried out at the subband level in the DWT domain by employing various smoothing masks. The lowest resolution tiles of the panorama are not treated while the mid-level resolution and the highest resolution tiles are modified to join them together.

## 7.3 The Proposed Joining Method

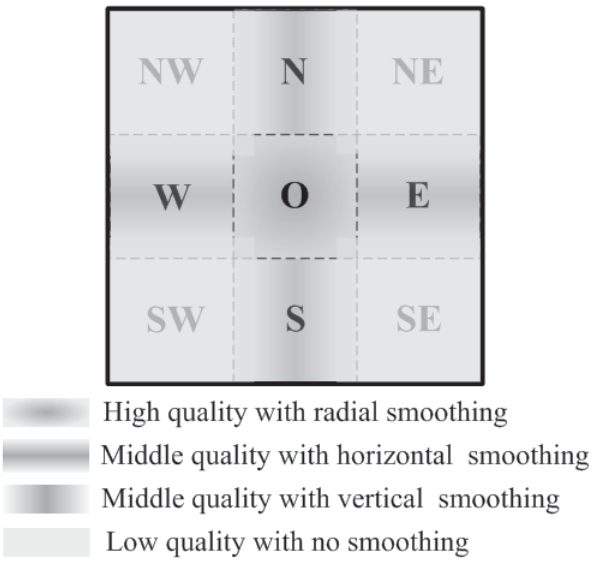



Figure 7.2: The tessellation scenario.

We are supposing a simplistic scenario of a panorama of $3 \times 3$ tiles, as depicted in Fig. 7.2. An individual tile is large enough to circumscribe the field of view even if the observer is too far. In other words one can say that if the observer is too far, his view is limited to a $3 \times 3$ tessellation at maximum with all the tiles at their lowest resolutions but as soon as he draws closer, further detail is added to the tile of focus to improve its resolution. The smoothing process will be done only on a $3 \times 3$ tiles. The rest of the encompassing tiles are in the low resolution and hence there is no need to smooth them. The tessellation is heterogeneous in terms of resolution and a given tile can have one of the three qualities, namely high middle and low. The high quality resolution corresponds to the center which is the viewpoint focus. We are rendering the corners with low resolution and the 4-neighbors of the center with the middle resolution tiles. The terms high, low and middle are not static and their extent depends on the distance of observer from the view as well as the the time passed after the viewer changed his position. We are assuming this use case with three quality environment because it reflects most of the possible cases in our application. If we have more than 9 tiles, the same process can be applied between two neighboring tiles, i.e. the difference of one/two level in horizontal or vertical direction, and two to four levels in the diagonal direction. Thus the $3 \times 3$ tessellation is not binding; rather it represents a simplistic worst-case scenario wherein three different quality tiles can be seamlessly rendered. There may be a tessellation having concentric square rings with each successive ring having a different quality [Tanner 1998] but we believe that at any instant one would hardly be dealing with a rendering involving more than three resolutions. That is to say,

that our resolution set may contain a wide spectrum of resolutions but at a given time three of them would be rendered. Our strategy is aimed at the seamless 3D visualization as a function of the network connection and the computing resources of the client. The resolution of various tiles in the tessellation is a function of its distance $D$ between the observer and his/her focus on the terrain as explained in Section 6.2.2.For the LOD we are relying on the multiresolution nature of DWT from the JPEG2000 codec, as explained in the previous chapters. Let the high, the middle and the low resolution of our scenario correspond to $l_h$, $l_m$ and $l_l$ approximate images, respectively. The quality gap would essentially bring with it the popping artifacts at tile boundaries. To undo that, we are proposing a wavelet-domain context dependent smoothing strategy graphically explained in Fig. 7.2. At the corners (NW, NE, SW and SE) are the $l_l$-approximate images obtained from lowest $3(L - l_l) + 1$ subbands with the rest of $3l_l$ subbands being replaced by 0s. For the rest of the tiles one can partition their subbands into three sets. One, the lowest $3(L-l_l)+1$ subbands will remain untouched since they represent the lowest rendered quality. Two, the highest $3l_h$ subbands of center tile (O) and $3l_m$ subbands of its 4-neighbors (E,W,N and S) are stuffed with 0s. Three, the remaining $3(l_l - l_h)$ and $3(l_l - l_m)$ subbands of central tile and its 4-neighbors, respectively. It is this third set (or its subset thereof) which will be subjected to the DWT domain treatment with three different smoothing functions on each subband individually:

1. for the left/right neighbors (W and E) of center use the horizontal smoothing function,

2. for the above/below neighbors (N and S) of center use the vertical smoothing function,

3. for the center (O) use the radial smoothing function.

Note that we are affecting the quality of only a small part on the periphery of the tile during a given smoothing process. For any given subband, the smoothing function creates a scalar multiplication mask (over $[0, 1]$) of the size of the subband. The horizontal smoothing mask would have a central band of rows of 1's. The thickness of this band depends on the maximum diameter, $\delta$, of the view focus. If the subband belongs to the $k$th resolution level, then the thickness should be around $\delta/2^k$. Beyond the band of 1's the rows gradually attenuate on both sides approaching 0 at the periphery. Our horizontal smoothing function is described here in the form of algorithm 5. Along similar lines we can get a vertical smoothing mask which is nothing but a transpose of the horizontal mask for a given subband. The radial smoothing mask is a bit tricky. we would have a center core of 1's having a diameter of about $\delta/2^k$ for a the subband of the $k$th resolution level. The shape of this core is more or less circular - octagonal, to be precise. Each ring of coefficients around the core would gradually attenuate approaching 0 at the corners and around 0.5 in the up/down and left/right directions. Algorithm 6 is the logical representation of our radial smoothing.

**Algorithm 5** Horizontal smoothing function.

**begin**
**read** width(w) of a DWT domain square subband
**set** $\delta \longleftarrow w/2$
**set** $min\_coeff \longleftarrow 0$
**Initialize** a vector $B$ of size $w * w$ to 1.0
**for** $j = 0$ *to* $w/2 - \delta/2$ **do**
  **set** $min\_coeff \longleftarrow min\_coeff + 2/(h - \delta)$
  **for** $i = 0$ *to* $w$ **do**
    **set** $B[i + w * j] \longleftarrow min\_coeff$
**for** $i = 0$ *to* $w * w/2$ **do**
  **set** $B[w * w - 1 - i] \longleftarrow B[i]$
**return** $B$ horizontal $w \times w$ smoothing mask **end**

**Algorithm 6** Radial smoothing function.

**begin**
**read** width(w) of a DWT domain square subband
**set** $\delta \longleftarrow w/2$
**set** $interval \longleftarrow 2/w$
**set** $w \longleftarrow w/2$
**for** $i = w * h - 1$ *to* $0$ **do**
  **if** $i >= w * (w - \delta/2)$ **then**
    **if** $i\%w >= (w - \delta/2)$ **then**
      **set** $temp[i] \longleftarrow 1.0$
    **else**
      **set** $temp[i] \longleftarrow temp[i + 1] - interval$
  **else**
    **set** $temp[i] \longleftarrow temp[i + w] - interval$
**set** $w \longleftarrow w * 2$
**for** $i = 0$ *to* $w * w/2$ **do**
  **if** $i\%w < w/2$ **then**
    **set** $B[i] \longleftarrow temp[i\% \lfloor w/2 \rfloor + \lfloor i/w \rfloor * \lfloor w/2 \rfloor]$
  **else**
    **set** $B[i] \longleftarrow B[i - (2 * (i\% \lfloor w/2 \rfloor) + 1)]$
  **set** $B[w * w - 1 - i] \longleftarrow B[i]$
**return** $B$ as radial smoothing square mask of width w
**end**

After the smoothing treatment of a subset or all of the third group of subbands and inverse DWT one can have a seamless visualization of the type given in Fig. 7.2. In our application, we are using the same strategy for the DEM to avoid popping effect on elevation data.

## 7.4 Experimental Results

We have applied our smoothing strategy to a practical example of $3072 \times 3072$ texture tessellation composed of 9 tiles of size $1024 \times 1024$. The corresponding DEM tessellation at our disposal has a size of $96 \times 96$ with each of the nine tiles having $32 \times 32$ coefficients. The texture tiles were transformed at level-4 reversible DWT in a standard JPEG2000 encoder whereas the DEM tiles were subjected to reversible level-4 DWT simply. In our example visualization, we are going for the worst case for the corners by rendering level-4 approximate tiles which would mean a rendering with only 0.4% of the coefficients. For the centre we are taking highest possible quality, i.e. level-0. The 4-neighbors of center have the middle quality of level-2 approximation. The high quality gap between the three qualities is likely to make the tile boundaries conspicuous. A texture tessellation with these tiles without any smoothing treatment is shown in Fig. 7.3.a. The boundaries are not that visible due to the fact that the space we are using to render a large image of $3072 \times 3072$ pixels is too small. For this purpose we have magnified a small area (corresponding to the small square in Fig. 7.3.a) from the tessellation in Fig. 7.3.b. We have deliberately selected a region that contain each of the resolution and smooth quality. This magnification is only for a comparison and the magnified portion does not represent the visual effect we are upto, since the distance it represents warrant improved qualities at corners, in our strategy.

Let $d$ implies that $3d + 1$ lowest subbands have been avoided during smoothing, e.g. $d = 1$ implies that the lowest 4 subbands have not been touched. Based on $d$ we are defining $d_h$ which corresponds to the center tile and $d_m$ that corresponds to the 4-neighbors. Note that since the 4-neighbors are level-2 approximates, the highest six subbands are already zero. This implies that $d_m$ cannot exceed 1. If we do not tamper the lowest four subbands during the horizontal smoothing of E/W and vertical smoothing of N/W then $d_m = 1$, otherwise $d_m = 0$ where the only lowest level-4 subband is avoided from smoothing. $d_h$, which corresponds to the center O, can assume a value of up to 3 and it was observed that the border between the four tiles gradually smoothen as we decrease the value of $d_h$ since more and more subbands are being involved in the radial smoothing of O. Since we are working on a worst case scenario and we are closer than we should be, the borders does not disappear at all but in the original environment the transition is seamless. Due to space constraint we are showing only one example smoothing in Fig. 7.4.a, where $d_l = 0$ and $d_h = 1$. Its magnified part corresponding to Fig. 7.3.b is illustrated in Fig. 7.4.b. From the latter, it can be observed that the border between the four tiles is far more smoother than the original.

Along the same lines the DEM can also be treated but the example at our disposal has a $96 \times 96$ DEM for a panorama of $3072 \times 3072$ which we believe is too small to be efficiently decomposed. It would have been interesting to take into account a larger DEM but due to circumstances beyond our control we were

(a) The original without smoothing

(b) zoomed square of (a)

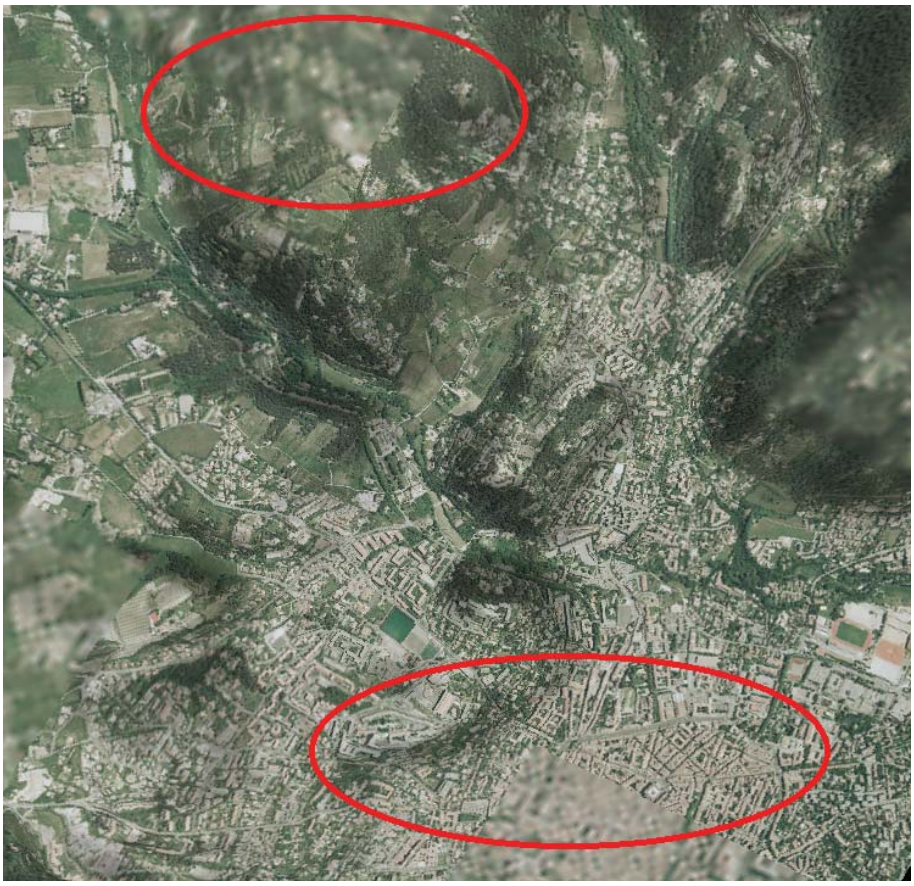
Figure 7.3: Example texture tessellation.

compelled to use a smaller DEM. Nowadays the DEM resolutions are approaching those of textures and hence for such DEMs one can safely apply the smoothing.

(a) Full panorama with $d_m = 0,\ d_h = 1$

(b) zoomed area of (c)

Figure 7.4: Example texture tessellation after smoothing: $l_h = 0$,$l_m = 2$,$l_l = 4$.



(a) No smoothing

(b) Smoothing: $d_m = 1,\ d_h = 3$

Figure 7.5: A 3D view corresponding to Fig. 7.3 and Fig. 7.4, respectively.

A 3D visualization realized from a subset portion, at a particular angle, of the original example and one of the smoothed version, with $d_l = 0$ and $d_h = 1$, are illustrated in Fig. 7.5. A 3D view with the original can be seen in Fig. 7.5.a., marked by red ovals for the sake of comparison. Fig. 7.5.b shows the 3D view of our smoothed example and a comparison of the areas inside the red ovals with those in Fig. 7.5.a. suggests that the borders between the tiles are more diluted in the former case.

## 7.5 Summary

The results have been interesting in the face of the fact that we presented a worst case example tessellation in terms of quality difference as well as DEM size [Hayat 2009d]. The question arises that if the DEM is different enough to warrant its own LOD, how can one synchronize it? The answer to this is the fact that since the LOD is wavelet driven, one can always synchronously unify the DEM texture pairs as explained in [Hayat 2008b] or even adapt the synchronization [Hayat 2008c] to exclude some subbands from embedding. In the continuation of our work we would like to add some other techniques to our strategy, the foremost being mosaicking. We believe that if we mix up our smoothing strategy with DWT domain blending of subbands in adjacent heterogeneous tiles, the smoothing must be improved further. Second in line may well be some more sophisticated function instead of the linear one that have a better conformance to the DWT process computation between multiple coefficients.

Chapter 8

# Conclusion and Perspective

*Dans ces travaux de thèse, nous avons adopté plusieurs approches afin de traiter le problème de visualisation 3D progressive dans un environnement client/serveur. D'une part, nous avons exploité la nature multi-résolution du codeur JPEG2000 et d'autre part, nous avons unifié les différents types de données dans un seul fichier standard en utilisant des techniques d'insertion de données cachées. Enfin, nous avons abordé le problème de raccordement de dalles décomposant une même image, mais à des résolutions différentes, sans soudure visible. Regroupées, toutes ces approches conduisent à un scénario de mise en place d'un système qui se veut évolutif, et offre une synchronisation pour la visualisation intéressante. Pour mettre en évidence l'efficacité de ce scénario, nous avons appliqué nos méthodes à des exemples pratiques dans les domaines de la visualisation de modèles numériques de terrains et de visages.*

*Dès les premières approches présentées, basées sur le codeur JPEG2000 sans perte avec insertion synchronisée des données 3D, nous constatons que les résultats obtenus sont intéressants. En prenant comme objectif principal l'imperceptibilité, nous constatons que même avec seulement* $0.1\%$ *à* $0.4\%$ *des coefficients, des visualisations satisfaisantes ont pu être réalisées. Ces résultats ont été fortement améliorés lorsque nous avons utilisé le mode avec pertes du codeur JPEG2000. Dans ce cas, afin d'améliorer la capacité d'insertion, des optimisations ont été proposées afin d'appliquer notre méthode à la visualisation de visages en 3D. Une stratégie de choix des composantes a été mise en place afin d'améliorer l'insertion de la carte de profondeur du fait de la forte probabilité de zéros dans les hautes fréquences.*

*Pour améliorer la qualité visuelle des différents approximations, une adaptation de la synchronisation a ensuite été proposée par insertion des données 3D dans un sous-ensemble des sous-bandes de la texture. Ceci signifie que la carte de profondeur a dû être décomposée différemment de la texture. De plus, pour des raisons de robustesse nous avons changé les techniques d'insertion en utilisant des approches par étalement de spectres. Cette nouvelle approche a montré une robustesse importante en regard d'ajout de bruit Gaussien et de la compression JPEG. Avec cette approche, nous avons également montré qu'il était possible de supprimer le message caché après extraction. Cette technique a cependant ses limites dans le sens que la taille de la carte de texture doit être nettement plus importante que la taille de la carte de profondeur.*

*Une fois ces stratégies mises en place, un effort a été fait afin de générer une transition douce entre plusieurs dalles jointives de différentes qualités. A partir de trois*

*qualités différentes pour les dalles visualisées, nous avons développé une approche de lissage dans le domaine des ondelettes. Nous avons été en mesure d'obtenir de bons résultats concernant ce problème de lissage. Cette technique a besoin d'améliorations futures, par exemple en utilisant des approches de mosaïques pour certaines sous-bandes dans le domaine des ondelettes.*
*De nombreuses pistes sont possibles pour améliorer ces travaux. Par exemple, l'étude de la robustesse est vraiment intéressante, en particulier dans le cas où nous souhaitons réaliser une visualisation 3D avec uniquement une partie découpée de l'image transmise codée. Les contraintes que nous avons eues doivent également être étudiées. En ce qui concerne la triangulation, des pistes sont possibles en utilisant une grille non-uniforme sur les différents niveaux de détails. Ceci permettrait ainsi une réduction considérable du nombre de triangles nécessaires à une bonne représentation du modèle 3D. Nous pensons également prétraiter les données de la carte de profondeur avant l'insertion. Nous avons présenté des premières améliorations concernant la capacité d'insertion, mais cet aspect peut encore être amélioré du fait que des coefficients nuls ou quasi nuls sont très fréquents dans les sous-bandes des hautes fréquences. Ainsi, une nouvelle distribution de probabilité, estimée a priori doit être étudiée. Dans un avenir proche, nous pensons appliquer le concept de "embedded zero-tree wavelets" (EZW) [Shapiro 1993] aux données avant l'insertion. Pour améliorer les cartes de profondeur, l'application de codage spécifique par régions d'intérêt doit également être étudiée. Enfin, dans un contexte d'une application complète, la gestion des flux entre un serveur et des clients doit être menée à bien.*

In this thesis, we adopted many approaches to deal with the problem of scalable 3D visualization in a client/server environment. On one hand we exploited the multiresolution nature of the JPEG2000 and on the other we unified the disparate information into a single standard file by employing data hiding techniques. Finally, we addressed the problem of seamless tessellation constituted by tiles of heterogeneous resolutions. Together, all these approaches lead to a scenario of putting in place a system that is scalable and offers interesting synchronized visualization. To highlight its efficacy we have applied our methods to practical examples from the visualization areas of face and digital elevation models.

Since the presentation of the first approaches, based on the lossless JPEG2000 codec with the synchronized unification of 3D data, the results have been interesting. With imperceptibility as a goal, we saw that even with only $0.1 - 0.4\%$ coefficients, satisfactory visualizations were realized. The bitrate tolerance was improved further when we changed the mode of JPEG2000 to lossy. In this case, in order to improve the embedding capacity, some optimizations were proposed in the case of the 3D visualization of facial data. A strategy, of choosing between various components, was established to optimize the depth map data in the DWT domain due to high probability of zeros in the high frequencies.

To improve the quality of the various approximation, an adaptation in synchronization was proposed by synchronizing with a subset of texture subbands. That would mean that depth map has to be decomposed differently than the texture. Side by side, the embedding technique was changed to spread spectrum for the sake of robustness. This new approach proved to be reasonably robust to JPEG compression and Gaussian noise addition. With this approach, we also showed that it was possible to remove the hidden message after the extraction. This technique, however, has its limits in the sense that the size of the texture map must be significantly larger than the size of the depth map.

Once the strategies in place, an effort was made to generate a soft transition between several neighboring tiles of different qualities. From three different qualities of visualization tiles, we have developed a DWT domain smoothing approach. We were able to get good results for this problem of smoothing. But the technique needs some improvement and the foremost in line is the use of mosaicking for selected subbands in the DWT domain.

Many prospects are there to improve this work. For example, the robustness study is really interesting, especially in the case of cropping wherein an effort was made to realize a 3D visualization with a small patch that is cropped from the encoded image. The constraints, we had, need further be investigated. As far as triangulation is concerned, there is also every likelihood of using a non-uniform grid on various levels of details. This will allow for a considerable reduction in the number of triangles necessary for a good representation of the 3D model. One would like to further preprocess the depth map data before embedding. We had applied a raw optimization but this aspect must be further refined in the face of the fact that zeros and near-zeros in low frequency subbands are usually replicated in higher frequency subbands. Thus a new probability distribution needs to be investigated, *a priori*. In

the near future we are mulling the application of the concept of embedded zero-tree wavelets (EZW [Shapiro 1993]) to the data before embedding. For improving the depth maps, the application of region of interest coding (ROI) must also be investigated. Lastly, in the context of a complete application,the streaming management between the server and its clients must also be given a serious thinking.

Appendix A

# List of Publications

## Journals

[1] K. Hayat, W. Puech, and G. Gesquière. Scalable 3D Visualization Through Reversible JPEG2000-Based BlindData Hiding. *IEEE Trans. Multimedia*, 10(7):1261–1276, November 2008.

[2] K. Hayat, W. Puech, and G. Gesquière. An Adaptive Spread Spectrum Synchronous Data Hiding Strategy for Scalable 3D Visualization. submitted, .

## Book Chapter

[1] K. Hayat, W. Puech, and G. Gesquière. *Recent Advances in Signal Processing*, chapter JPEG2000-Based Data Hiding and Applications. In-Tech, 2009. (to appear).

## Conferences

[1] 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. EUSIPCO'07: the 15th European Signal Processing Conference*, pages 2519–2523, Poznan, Poland, September 2007.

[2] K. Hayat, W. Puech, and G. Gesquière. A Lossy JPEG2000-Based Data Hiding Method for Scalable 3D Terrain Visualization. In *Proc. EUSIPCO'08: the 16th European Signal Processing Conference*, Lausanne, Switzerland, August 2008.

[3] K. Hayat, W. Puech, and G. Gesquière. Scalable Data Hiding for Online Textured 3D Terrain Visualization. In *Proc. ICME'08, IEEE International Conference on Multimedia & Expo*, pages 217–220, June 2008.

[4] K. Hayat, W. Puech, and G. Gesquière. An Adaptive Spread Spectrum (SS) Synchronous Data Hiding Strategy for Scalable 3D Terrain Visualization. In *Proc. SPIE, Electronic Imaging, Visualization and Data Analysis*, volume 7243, San Jose, CA, USA, January 2009.

[5] K. Hayat, W. Puech, and G. Gesquière. Seamless Joining of Tiles of Varying Resolutions for Online 3D Terrain Visualization by DWT Domain Smoothing. In *EUSIPCO'09: the 17th European Signal Processing Conference*, Glasgow, Scotland, August 2009.

[6] 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*, volume 6495, pages 64950N(1–10), San Jose, CA, USA, January 2007.

[7] K. Hayat, W. Puech, G. Gesquière, and G. Subsol. 3D Facial Visualization through Adaptive Spread Spectrum Synchronous Scalable (A4S) Data Hiding. In *Proc. IEEE International Conference on Image Processing, ICIP09*, Cairo, Egypt, November 2009.

[8] K. Hayat, W. Puech, G. Gesquière, and G. Subsol. JPEG2000-Based Data Hiding to Synchronously Unify Disparate Facial Data for Scalable 3D Visualization. In *Singaporean-French Symposium, SinFRa'09: Image, Perception, Access & Language Laboratory (IPAL)*, Fusionopolis, Singapore, Febuary 2009.

## French Language Conferences

[1] K. Hayat, W. Puech, and G. Gesquière. Une méthode efficace d'insertion de données cachées basée sur JPEG2000 sans perte pour une visualisation scalable et synchronisée de terrain 3D. In *Proc. $XXI^e$ Colloque GRETSI, Traitement du Signal et des Images*, Troyes, France, Septembre 2007.

[2] K. Hayat, W. Puech, G. Gesquière, and M. Chaumont. Visualisation 3D temps-réel à distance de MNT par insertion de données cachées basée ondelettes. In *Proc. CORESA2006, COmpression et REprésentation des Signaux Audiovisuels*, Caen, France, Novembre 2006.

# Bibliography

[Abdul-Rahman 2008] A. Abdul-Rahman and M. Pilouk. Spatial Data Modelling for 3D GIS. Springer, 2008. vii, 33, 34

[Adams 2005] M. D. Adams. The JPEG-2000 Still Image Compression Standard (Last Revised: 2005-12-03). www.ece.uvic.ca/∼mdadams, 2005. 20, 21

[Agreste 2007] S. Agreste, G. Andaloro, D. Prestipino and L. Puccio. *An Image Adaptive, Wavelet-Based Watermarking of Digital Images.* Journal of Computational and Applied Mathematics, vol. 210, no. 1–2, pages 13–21, 2007. 63

[Aslantas 2008] V. Aslantas, A .L. Dogan and S. Ozturk. *DWT-SVD Based Image Watermarking Using Particle Swarm Optimizer.* In Proc. ICME'08, IEEE International Conference on Multimedia & Expo, pages 241–244, June 2008. 63

[Balter 2006] R. Balter, P. Gioia and L. Morin. *Scalable and Efficient Coding Using 3D Modeling.* IEEE Trans. Multimedia, vol. 8, no. 6, pages 1147–1155, December 2006. 38

[Baumann 1997] P. Baumann, P. Furtado, R. Ritsch and N. Widmann. *Geo/Environmental and Medical Data Management in the RasDaMan System.* In Proc. 23rd International Conference on Very Large Data Bases (VLDB'97), pages 548–552, Athens, Greece, August 1997. 41

[Bechmann 2007] D. Bechmann and B. Péroche. Informatique graphique, modélisation géométrique et animation. Hermès - Lavoisier, February 2007. 37

[Bender 1996] W. Bender, D. Gruhl, N. Morimoto and A. Lu. *Techniques for Data Hiding.* IBM Systems Journal, vol. 35, no. 3-4, pages 313–336, February 1996. 51

[Bishop 1998] L. Bishop, D. Eberly, T. Whitted, M. Finch and M. Shantz. *Designing a PC Game Engine.* IEEE Computer Graphics and Applications, vol. 18, no. 1, pages 46–53, 1998. 41

[Bowyer 2006] K. W. Bowyer, K. Chang and P. Flynn. *A Survey of Approaches and Challenges in 3D and Multi-modal 3D + 2D Face Recognition.* Computer Vision & Image Understanding, vol. 101, no. 1, pages 1–15, 2006. vii, 38, 45

[Buchholz 2005] H. Buchholz and J. Döllner. *View-Dependent Rendering of Multiresolution Texture-Atlases.* In Proc. IEEE Visualization 2005, pages 215–222, 2005. 44, 45

[Chaumont 2007] M. Chaumont and W. Puech. *3D Face Model Tracking Based on a Multiresolution Active Search.* In Proc. SPIE, Electronic Imaging, Visual Communications and Image Processing, volume 6508, pages 65081U.1–65081U.8, San Jose, CA, USA, January 2007. 46

[Chauve 2007] A. Chauve, F. Bretar, S. Durrieu, M. Pierrot-Deseilligny and W. Puech. *Processing Full-Waveform Lidar Data: Modelling Raw Signals.* In Proc. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, volume 36, pages 102–107, Espoo, Finland, 2007. 106

[Chui 1992] C. K. Chui. An Introduction to Wavelets. Academic Press Professional, Inc., San Diego, CA, USA, 1992. 19

[Conde 2005] C. Conde and A. Serrano. *3D Facial Normalization with Spin Images and Influence of Range Data Calculation over Face Verification.* In Proc. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 16, pages 115–120, June 2005. 38, 46

[Cox 2008] I. J. Cox, M. L. Miller and J. A. Bloom. Digital Watermarking. Morgan Kaufmann Publishers, 2008. 54, 56, 59, 105

[Danovaro 2006] E. Danovaro, L. De Floriani, P. Magillo, E. Puppo and D. Sobrero. *Level-of-detail for Data Analysis and Exploration: A Historical Overview and some New Perspectives.* Computers & Graphics, vol. 30, no. 3, pages 334 – 344, 2006. 45

[Daubechies 1992] I. Daubechies. Ten Lectures on Wavelets. SIAM, Philadelphia, PA, 1992. 17, 19

[Daubechies 1998] I. Daubechies and W. Sweldens. *Factoring Wavelet Transforms into Lifting Steps.* Fourier Anal. Appl., vol. 4, no. 3, 1998. 23, 76, 90

[Dawei 2004] Z. Dawei, C. Guanrong and L. Wenbo. *A Chaos-Based Robust Wavelet-Domain Watermarking Algorithm.* Chaos, Solitons & Fractals, vol. 22, no. 1, pages 47–54, 2004. 63

[Deb 2006] S. Deb, S. Bhattacharjee, S. Patidar and P. J. Narayanan. *Real-time Streaming and Rendering of Terrains.* In Computer Vision, Graphics and Image Processing, 5th Indian Conference, ICVGIP, pages 276–288, Madurai, India, 2006. 44

[Döllner 2000] J. Döllner, K. Baumann and K. Hinrichs. *Texturing Techniques for Terrain Visualization.* In Proc. the 11th IEEE Visualization 2000Conference (VIS 2000), Washington, DC, USA, 2000. 44

[Dugelay 1999] J.L. Dugelay, K. Fintzel, S. Valente and H. Delingette. *Clonage de visage et spatialisation video : outils pour la téléconférence virtuelle.* Traitement du Signal, vol. 16, no. 1, pages 61–72, 1999. 46

[Farin 2002] G. Farin. Curves and Surfaces for CAGD: a Practical Guide. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002. 35

[Floriani 1995] L. De Floriani and E. Puppo. *Hierarchical Triangulation for Multiresolution Surface Description.* ACM Trans. Graph, vol. 14, no. 4, pages 363–411, 1995. 41

[Fowler 1979] R. J. Fowler and J. J. Little. *Automatic Extraction of Irregular Network Digital Terrain Models.* In Proc. SIGGRAPH '79: the 6th Annual Conference on Computer Graphics and Interactive Techniques, pages 199–207, New York, NY, USA, 1979. 41

[Fridrich 2003] J. Fridrich and M. Goljan. *Digital Image Steganography using Stochastic Modulation.* In Proc. SPIE, Electronic Imaging, pages 191–202, Santa Clara, CA, USA, January 2003. 54

[Frueh 2004] C. Frueh, R. Sammon and A. Zakhor. *Automated Texture Mapping of 3D City Models With Oblique Aerial Imagery.* In Proc. 3DPVT '04: the 2nd International Symposium on 3D Data Processing, Visualization and Transmission, pages 396–403, Washington, DC, USA, 2004. 44

[Gerlek 2004] 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. 43

[Gioia 2004] P. Gioia, O. Aubaut and C. Bouville. *Real-Time Reconstruction of Wavelet Encoded Meshes for View-Dependent Transmission and Visualization.* IEEE Trans. Circuits and Systems for Video Technology , vol. 14, no. 7, pages 1009–1020, July 2004. 43

[Hansen 1999] P. Hansen. *OpenGL Texture-Mapping with Very Large Datasets and Multi-Resolution Tiles.* In Proc. SIGGRAPH'99: the 26th Annual Conference on Computer Graphics and Interactive Techniques, page 262, New York, NY, USA, 1999. 41

[Hayat 2006] K. Hayat, W. Puech, G. Gesquière and M. Chaumont. *Visualisation 3D temps-réel à distance de MNT par insertion de données cachées basée ondelettes.* In Proc. CORESA2006: COmpression et REprésentation des Signaux Audiovisuels, Caen, France, Novembre 2006. 75

[Hayat 2007a] 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. EUSIPCO'07: the 15th European Signal Processing Conference, pages 2519–2523, Poznan, Poland, September 2007. 88

[Hayat 2007b] K. Hayat, W. Puech and G. Gesquière. *Une méthode efficace d'insertion de données cachées basée sur JPEG2000 sans perte pour une visualisation scalable et synchronisée de terrain 3D.* In Proc. $XXI^e$ Colloque GRETSI, Traitement du Signal et des Images, Troyes, France, Septembre 2007. 88

[Hayat 2007c] 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, volume 6495, pages 64950N(1–10), San Jose, CA, USA, January 2007. 75

[Hayat 2008a] K. Hayat, W. Puech and G. Gesquière. *A Lossy JPEG2000-Based Data Hiding Method for Scalable 3D Terrain Visualization.* In Proc. EUSIPCO'08: the 16th European Signal Processing Conference, Lausanne, Switzerland, August 2008. 95

[Hayat 2008b] K. Hayat, W. Puech and G. Gesquière. *Scalable 3D Visualization Through Reversible JPEG2000-Based Blind Data Hiding.* IEEE Trans. Multimedia, vol. 10, no. 7, pages 1261–1276, November 2008. 88, 129, 137

[Hayat 2008c] K. Hayat, W. Puech and G. Gesquière. *Scalable Data Hiding for Online Textured 3D Terrain Visualization.* In Proc. ICME'08, IEEE International Conference on Multimedia & Expo, pages 217–220, June 2008. 125, 137

[Hayat 2009a] K. Hayat, W. Puech and G. Gesquière. *An Adaptive Spread Spectrum (SS) Synchronous Data Hiding Strategy for Scalable 3D Terrain Visualization.* In Proc. SPIE, Electronic Imaging, Visualization and Data Analysis, volume 7243, San Jose, CA, USA, January 2009. 114

[Hayat 2009b] K. Hayat, W. Puech and G. Gesquière. An Adaptive Spread Spectrum Synchronous Data Hiding Strategy for Scalable 3D Visualization. (submitted), 2009. 125

[Hayat 2009c] K. Hayat, W. Puech and G. Gesquière. Recent Advances in Signal Processing, chapitre JPEG2000-Based Data Hiding and Applications. In-Tech, 2009. (to appear). 59

[Hayat 2009d] K. Hayat, W. Puech and G. Gesquière. *Seamless Joining of Tiles of Varying Resolutions for Online 3D Terrain Visualization by DWT Domain Smoothing.* In EUSIPCO'09: the 17th European Signal Processing Conference, Glasgow, Scotland, August 2009. 137

[Hayat 2009e] K. Hayat, W. Puech, G. Gesquière and G. Subsol. *JPEG2000-Based Data Hiding to Synchronously Unify Disparate Facial Data for Scalable 3D Visualization.* In Singaporean-French Symposium, SinFRa'09: Image, Perception, Access & Language Laboratory (IPAL), Fusionopolis, Singapore, Febuary 2009. 101

[Heurtebise 2006] X. Heurtebise, S. Thon and G. Gesquière. *Multiresolution Representation and Deformation of Wavelet-Based 3D Objects*. In Proc. International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision (Winter School of Computer Graphics - WSCG'06), Plzen - Bory, Czech Republic, 2006. 35

[Hoppe 1996] H. Hoppe. *Progressive Meshes*. In Proc. SIGGRAPH'96: the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pages 99–108, New Orleans, LA, USA, 1996. 41

[Hwa 2004] L. M. Hwa, M. A. Duchaineau and K. I. Joy. *Adaptive 4-8 Texture Hierarchies*. In Proc. VIS '04: the Conference on Visualization '04, pages 219–226, Washington, DC, USA, 2004. 45

[Inoue 1998] H. Inoue, A. Miyazaki, A. Yamamoto and T. Katsura. *A Digital Watermark Based on the Wavelet Transform and its Robustness on Image Compression*. In Proc. ICIP 98. the 1998 International Conference on Image Processing, volume 2, pages 391–395 vol.2, October 1998. 63

[Ishida 2008] T. Ishida, K. Yamawaki, H. Noda and M. Niimi. *Performance Improvement of JPEG2000 Steganography Using QIM*. In Proc. IIH-MSP '08: the 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pages 155–158, Washington, DC, USA, 2008. 65

[ISO/IEC 2000] ISO/IEC. ISO/IEC 14492-1: Lossy/Lossless Coding of Bi-Level Images. ISO Central Secretariat: CH-1211 Geneva, Switzerland, 2000. 24

[ISO/IEC 2004] 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. 20, 21

[Johnson 1998] N. F. Johnson and S. Jajodia. *Steganalysis: The Investigation of Hidden Information*. In Proc. the IEEE Information Technology Conference, pages 113–116, 1998. 53

[Kahn 1967] D. Kahn. The Codebreakers. Macmillan, 1967. 49, 50

[Kaufman 1993] A. Kaufman, D. Cohen and R. Yagel. *Volume Graphics*. IEEE Computer, vol. 26, no. 7, pages 51–64, July 1993. 30, 35

[Kerckhoffs 1883] A. Kerckhoffs. *La cryptographie militaire*. Journal des sciences militaires, vol. 9, pages 5–38, 1883. 50

[Kharrazi 2006] M. Kharrazi, H. T. Sencar and N. Memon. *Performance Study of Common image Steganography and Steganalysis Techniques*. Journal of Electronic Imaging, vol. 15, no. 4, page 041104, 2006. 66

[Kim 2004] J. K. Kim and J. B. Ra. *A Real-Time Terrain Visualization Algorithm Using Wavelet-Based Compression.* The Visual Computer, vol. 20, no. 2–3, pages 67–85, 2004. 111

[Kong 2004] X. Kong, Y. Liu, H. Liu and D. Yang. *Object Watermarks for Digital Images and Video.* Image and Vision Computing, vol. 22, pages 583–595, 2004. 62

[Kundur 1997] 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), volume 1, pages 544–547, Santa Barbara, CA, USA, October 1997. 62

[Kundur 1998] D. Kundur and D. Hatzinakos. *Digital Watermarking Using Multiresolution Wavelet Decomposition.* In Proc. IEEE International Conference on Acoustic, Speech and Signal Processing (IEEE ICASSP 98), volume 5, pages 2969–2972, Seattle, WA, USA., May 1998. 61

[Kundur 1999] D. Kundur. *Improved Digital Watermarking Through Diversity and Attack Characterization.* In Proc. ACM Workshop on Multimedia Security'99, pages 53–58, Orlando, FL, USA, October 1999. 61

[Lafarge 2008] F. Lafarge, X. Descombes, J. Zerubia and M. Pierrot-Deseilligny. *Building Reconstruction from a Single DEM.* In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1–8, June 2008. 124

[Lake 2005] R. Lake, D. Burggraf, M. Kyle and S. Forde. GML in JPEG 2000 for Geographic Imagery (GMLJP2) Implementation Specification. Numeéro OGC 05-047r2. Open Geospatial Consortium (OGC), 2005. 43

[Lakhia 2004] A. Lakhia. *Efficient Interactive Rendering of Detailed Models with Hierarchical Levels of Detail.* In Proc. 3DPVT '04: the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium, pages 275–282, Washington, DC, USA, 2004. 45

[Larsen 2003] B.D. Larsen and N.J. Christensen. *Real-time Terrain Rendering using Smooth Hardware Optimized Level of Detail.* Journal of WSCG, vol. 11, no. 2, pages 282–9, February 2003. WSCG'2003: 11th International Conference in Central Europe on Computer Graphics, Visualization and Digital Interactive Media. 44

[Lenti 2000] J. Lenti. *Steganographic Methods.* Periodica Polytechnica Ser. Electrical Engineering, vol. 44, no. 3–4, pages 249–258, 2000. vii, 54, 55, 57, 59

[Li 1994] R. X. Li. *Data Structures and Application Issues in 3-D Geographic Information Systems.* Geomatica, pages 209–224, 1994. 32, 33

[Li 2003] K. Li and X. P. Zhang. *Reliable Adaptive Watermarking Scheme Integrated with JPEG2000.* In Proc. of 3rd International Symposium on Image and Signal Processing and Analysis (ISPA 2003), Rome, Italy, September 2003. 65

[Li 2004] B. Y. Li, H. S. Liao, C. H. Chang and S. L. Chu. *Visualization for HPC Data - Large Terrain Model.* In Proc. HPCASIA '04: the Seventh International Conference on High Performance Computing and Grid in Asia Pacific Region, pages 280–284, Washington, DC, USA, 2004. 44

[Lin 1999] E. T. Lin and E. J. Delp. *A Review of Data Hiding in Digital Images.* In Proc. the Image Processing, Image Quality, Image Capture Systems Conference (PICS '99), pages 274–278, 1999. 51, 54, 59

[Lin 2008] W.-H. Lin, S.-J. Horng, T.-W. Kao, P. Fan, C.-L. Lee and Y. Pan. *An Efficient Watermarking Method Based on Significant Difference of Wavelet Coefficient Quantization.* IEEE Trans. Multimedia, vol. 10, no. 5, pages 746–757, 2008. 65

[Liu 2006] 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, pages 356–367, 2006. 62

[Losasso 2004] F. Losasso and H. Hoppe. *Geometry Clipmaps: Terrain Rendering Using Nested Regular Grids.* ACM Trans. Graph., vol. 23, no. 3, pages 769–776, 2004. 41, 44, 128

[Luebke 2002] D. Luebke, M. Reddy, J. D. Cohen, A. Varshney, B. Watson and R. Huebner. Level of Detail for 3D Graphics. Morgan Kaufmann Publishers, San Francisco, CA, USA, July 2002. 107

[Maitya 2007] S. P. Maitya, M. K. Kundub and T. S. Das. *Robust SS Watermarking with Improved Capacity.* Pattern Recognition Letters, vol. 28, no. 3, pages 350–356, 2007. 63

[Mallat 1998] S. Mallat. A Wavelet Tour of Signal Processing. Academic Press, 1998. 19, 76, 90

[Marvel 1999] L.M. Marvel, Jr. Boncelet C.G. and C.T. Retter. *Spread Spectrum Image Steganography.* IEEE Trans. Image Processing, vol. 8, no. 8, pages 1075–1083, August 1999. 57

[Meerwald 2001a] P. Meerwald. Digital Image Watermarking in the Wavelet Transform Domain. Master's thesis, Department of Scientific Computing, University of Salzburg, Austria, January 2001. 54, 61

[Meerwald 2001b] 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, pages 69–79, May 2001. 66

[Meerwald 2001c] P. Meerwald and A. Uhl. *A Survey of Wavelet-Domain Watermarking Algorithms.* In Proc. SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents III, volume 4314, pages 505–516, San Jose, CA, USA, January 2001. 61

[Noda 2003] H. Noda, J. Spaulding, M. N. Shirazi, M. Niimi and E. Kawaguchi. *Bit-Plane Decomposition Steganography Combined with JPEG2000 Compression.* In IH '02: Revised Papers from the 5th International Workshop on Information Hiding, pages 295–309, London, UK, 2003. 66

[Noore 2007] 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, pages 188–194, 2007. 62

[Ohyama 2008] S. Ohyama, M. Niimi, K. Yamawaki and H. Noda. *Reversible Data Hiding of Full Color JPEG2000 Compressed Bit-Stream Preserving Bit-Depth Information.* In Proc. ICPR'08: the 19th International Conference on Pattern Recognition, pages 1–4, December 2008. 65

[Okamoto 2008] R. M. Okamoto, F. L. de Mello and C. Esperança. *Texture Management in View Dependent Application for Large 3D Terrain Visualization.* In Proc. SpringSim '08: the 2008 Spring Simulation Multiconference, pages 641–647, New York, NY, USA, 2008. 44

[Piva 2005] A. Piva, F. Bartolini and R. Caldelli. *Self Recovery Authentication of Images in the DWT Domain.* Int. J. Image Graphics, vol. 5, no. 1, pages 149–166, 2005. 64

[Puech 2001] W. Puech, A. G. Bors, I. Pitas and J. M. Chassery. *Projection Distortion Analysis for Flattened Image Mosaicing from Straight Uniform Generalized Cylinders.* Pattern Recognition, vol. 34, no. 8, pages 1657–1670, 2001. 108

[Pulli 1997] K. Pulli, M. Cohen, T. Duchamp and W. Stuetzle. *View-Based Rendering: Visualizing Real Objects from Scanned Range and Color Data.* In Eurographics Rendering Workshop, pages 23–34, 1997. 38

[Rodrigues 2004] J. M. Rodrigues, J. R. Rios and W. Puech. *A Review of Data Hiding in Digital Images.* In Proc. 5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'04, Lisbon, Portugal, 2004. 55

[Roudet 2009] C. Roudet, F. Dupont and A. Baskurt. *Semi-Regular 3D Mesh Progressive Compression and Transmission based on an Adaptive Wavelet Decomposition.* In Proc. SPIE Wavelet Applications in Industrial Processing VI, volume 7248, San Jose, CA, USA, January 2009. 37

[Royan 2007] J. Royan, P. Gioia, R. Cavagna and C. Bouville. *Network-Based Visualization of 3D Landscapes and City Models*. IEEE Computer Graphics and Applications, vol. 27, no. 6, pages 70–79, November 2007. 43

[Said 1996] A. Said and W.A. Pearlman. *A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees*. Circuits and Systems for Video Technology, IEEE Transactions on, vol. 6, no. 3, pages 243–250, June 1996. 63

[Schlauweg 2006] M. Schlauweg, D. Pröfrock and E. Müller. *JPEG2000-Based Secure Image Authentication*. In Proc. MM&Sec '06: the 8th Workshop on Multimedia and Security, pages 62–67, New York, NY, USA, 2006. 64

[Sederberg 1998] T. W. Sederberg, J. Zheng, D. Sewell and M. Sabin. *Non-Uniform Recursive Subdivision Surfaces*. In Proc. SIGGRAPH '98: the 25th Annual Conference on Computer Graphics and Interactive Techniques, pages 387–394, New York, NY, USA, 1998. 37

[Shapiro 1993] J.M. Shapiro. *Embedded Image Coding using Zerotrees of Wavelet Coefficients*. IEEE Trans. Signal Processing, vol. 41, no. 12, pages 3445–3462, 1993. 63, 140, 142

[Simmons 1984] G. J. Simmons. *The Prisoners' Problem and the Subliminal Channel*. In Proc. CRYPTO'83: Advances in Cryptology, pages 364–378, 1984. 49, 50

[Smith 1997] G. Smith and D.A. Atchison. The Eye and Visual Optical Instruments. Cambridge University Press, New York, USA, 1997. 107

[Su 2001] 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, pages 35–53, June 2001. 65

[Su 2003] P. C. Su and C .-C .J. Kuo. *Steganography in JPEG2000 Compressed Images*. IEEE Trans. Consumer Electronics, vol. 49, no. 4, pages 824–832, November 2003. 66

[Suhail 2003] M. A. Suhail, M. S. Obaidat, S. S. Ipson and B. Sadoun. *A Comparative Study of Digital Watermarking in JPEG and JPEG 2000 Environments*. Inf. Sci. Inf. Comput. Sci., vol. 151, pages 93–105, 2003. 62

[Sullivan 2004] K. Sullivan, Z. Bi, U. Madhow, S. Chandrasekaran and B.S. Manjunath. *Steganalysis of quantization index modulation data hiding*. In IEEE International Conference on Image Processing, pages 1165–1168, October 2004. 65

[Sweldens 1995] W. Sweldens. *The Lifting Scheme: a New Philosophy in Biorthogonal Wavelet Constructions.* In Proc. SPIE, Electronic Imaging, Wavelet Applications in Signal and Image Processing, volume 2569, pages 68–79, San Jose, CA, USA, September 1995. 19

[Tanner 1998] C. C. Tanner, C. J. Migdal and M. T. Jones. *The Clipmap: A Virtual Mipmap.* In Proc. SIGGRAPH '98: the 25th Annual Conference on Computer Graphics and Interactive Techniques, pages 151–158, New York, NY, USA, 1998. 44, 130

[Taubman 2002] D. S. Taubman and M. W. Marcellin. JPEG2000: Image Compression Fundamentals; Standards and Practice. Springer, 2002. vii, 20, 25, 26

[Tsai 2008] F. Tsai and H. C. Chiu. *Adaptive Level of Detail for Large Terrain Visualization.* In Proc. the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, ISPRS Congress Beijing 2008 (Commission IV), volume XXXVII, pages 579–584, October 2008. 44

[Uccheddu 2004] F. Uccheddu, M. Corsini and M. Barni. *Wavelet-Based Blind Watermarking of 3D Models.* In Proc. MM&Sec '04: Workshop on Multimedia and Security, pages 143–154, New York, NY, USA, 2004. 63

[Ueng 2000] M. F. Ueng and J. H. Chuang. *Terrain Rendering with View-Dependent LOD Caching.* In Tenth International Conference on Artificial Reality and Tele-existence (ICAT2000), October 2000. 44

[Vatsa 2009] M. Vatsa, R. Singh and A. Noore. *Feature Based RDWT Watermarking for Multimodal Biometric System.* Image and Vision Computing, vol. 27, no. 3, pages 293–304, 2009. 64

[Vepakomma 2008] U. Vepakomma, B. St-Onge and D. Kneeshaw. *Spatially Explicit Characterization of Boreal Forest Gap Dynamics Using Multi-Temporal Lidar Data.* Remote Sensing of Environment, vol. 112, no. 5, pages 2326–2340, 2008. 106

[Vetterli 1995] M. Vetterli and J. Kovačevic. Wavelets and Subband Coding. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995. 12

[Wagner 2003] D. Wagner. ShaderX2: Shader Programming Tips and Tricks with DirectX 9.0, chapitre Terrain Geomorphing in Vertex Shader. Wordware Publishing Inc., Plano, TX, USA, 2003. 44

[Wagner 2004] D. Wagner. Terrain Geomorphing in the Vertex Shader. Academic Press, 2004. 41

[Wang 1998] 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), pages 3271–3275, Seattle, WA, USA, May 1998. 61

[Wang 2004] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli. *Image Quality Assessment: from Error Visibility to Structural Similarity.* IEEE Trans. Image Processing, vol. 13, no. 4, pages 600–612, April 2004. 54

[Weik 1998] S. Weik, J. Wingbermuhle and W. Niem. *Automatic Creation of Flexible Antropomorphic Models for 3D Videoconferencing.* In Proc. CGI'98: Computer Graphics International, pages 520–527, 1998. 46

[Wloka 2004] M. Wloka. ShaderX3: Advanced Rendering with DirectX and OpenGL, chapitre Improved Batching via Texture Atlases. Charles River Media Inc., Hingham, Massachusetts, USA, 2004. 44

[Woo 2005] C. S. Woo, J. Du and B. Pham. *Performance Factors Analysis of a Wavelet-Based Watermarking Method.* In Proc. ACSW Frontiers '05: the Australasian Workshop on Grid Computing and e-Research, pages 89–97, Darlinghurst, Australia, 2005. 62

[Xia 1997] 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), pages 548–551, Santa Barbara, CA, USA, October 1997. 62

[Xiang 2007] S. Xiang and H. J. Kim. *Geometrically Invariant Image Watermarking in the DWT Domain.* In WISA, volume 4867 of *Lecture Notes in Computer Science*, pages 76–90, 2007. 62

[Yavuz 2007] E. Yavuz and Z. Telatar. *Improved SVD-DWT Based Digital Image Watermarking Against Watermark Ambiguity.* In Proc. SAC '07: the ACM Symposium on Applied Computing, pages 1051–1055, New York, NY, USA, 2007. 62

[Yen 2008] E. Yen and K. S. Tsai. *HDWT-Based Grayscale Watermark for Copyright Protection.* Expert Systems with Applications, vol. 35, no. 1–2, pages 301–306, 2008. 64

[Yilmaz 2008] A. Yilmaz and A. A. Alatan. *Error Detection and Concealment for Video Transmission Using Information Hiding.* Image Commun., vol. 23, no. 4, pages 298–312, 2008. 52

[Yin 2001] K. Yin, Z. Pan, J. Shi and D. Zhang. *Robust Mesh Watermarking Based on Multiresolution Processing.* Computers & Graphics, vol. 25, no. 3, pages 409–420, 2001. 64

[Yu 1996] J. Yu and D. J. DeWitt. *Processing Satellite Images on Tertiary Storage: A Study of the Impact of Tile Size on Performance.* In Proc. 5th NASA Goddard Conference on Mass Storage Systems and Technologies, 1996. 41

[Yu 2003] Z. Yu, H. H. S. Ip and L. F. Kwok. *A Robust Watermarking Scheme for 3D Triangular Mesh Models.* Pattern Recognition, vol. 36, no. 11, pages 2603–2614, 2003. 64

**Abstract:** The principal objective of this thesis is to unify disparate 3D information and then realize scalable visualization in a client/server environment that is heterogeneous in terms of network, computing and memory resources. For scalability we are exploiting the multiresolution nature of the discrete wavelet transform (DWT) from the state of the art JPEG2000 codec. The data unification is being carried out through DWT domain blind data hiding that may either be fully or adaptively synchronous. A typical surface based 3D visualization requires at least two sets of data: a 2D intensity image, called texture, with a corresponding 3D shape rendered in the form of a range image, a shaded 3D model and/or 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. The texture is a corresponding 2D color image which is overlaid onto a model produced from the depth map by triangulation. For data hiding, the range data is first subjected to DWT while the texture data is input to the JPEG2000 encoder. The JPEG2000 coding pipeline of texture is interrupted at some stage after the DWT step and the DWT domain range coefficients are embedded in the all or a subset of texture subbands. The embedded data is reintroduced to the JPEG2000 pipeline at the same step where the interruption was made. Since the JPEG2000 format is conserved during the process, the resultant code can be sent across any communication channel like any other JPEG2000 file. The resolution scalability of wavelets and the synchronized character of our techniques enable a 3D visualization even with fewer than original resolution bands 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. In the first phase of this work the thrust was on the perceptual transparency and that is why least significant bit (LSB) embedding was employed for both the synchronous and adaptively synchronous cases. The second phase concerns robustness and that is a why spread spectrum (SS) strategy is utilized in embedding. During the latter phase the imperceptibility has not been ignored at all as the embedding is removable. Examples from the two main areas of terrain and face visualization have been taken as case studies to prove the utility of our methods. In the third and final phase we take a broader aspect of the problem when we try to render the underlying tile components of a heterogeneous tessellation, seamlessly. This seamless joining of tiles is ensured through special DWT domain smoothing functions.

**Keywords:** 3D visualization, scalability, synchronization, adaptive, data hiding, embedding, LSB, spread spectrum, multiresolution, DWT, JPEG2000, subband, DEM, range data, 2.5D, texture, depth map, terrain, face

**Résumé :** L'objectif principal de ces travaux de thèse est d'unifier différentes informations 2D et 3D afin de réaliser une visualisation adaptée dans un environnement client/serveur hétérogène en termes de réseau, de traitement et de ressources mémoire. Dans ce contexte, nous avons exploité la nature multi-résolution de la transformée en ondelettes discrètes (TOD) du codeur JPEG2000. L'unification des données est réalisée par insertion aveugle, synchrone ou partiellement synchrone, des données cachées dans le domaine des ondelettes. Une visualisation 3D classique nécessite au moins deux types de données : une image 2D d'intensité, appelé texture, et une forme 3D pouvant être représentée par une image, un modèle 3D ombré ou un maillage de points. Ce type d'image, parfois également appelé carte de profondeur est une image dans laquelle la valeur des pixels reflète la distance du capteur à la surface par imagerie. La texture est une image 2D couleur qui est apposée sur le modèle 3D après triangulation. Au niveau de l'insertion des données cachées, la carte de profondeur est d'abord transformée dans le domaine des ondelettes tandis que la texture est insérée dans le codeur JPEG2000. Le processus de codage JPEG2000 de la texture est interrompue, et les coefficients 3D sont insérés dans la totalité ou dans un sous-ensemble des sous-bandes de la texture. Les données sont re-intégrées dans le codeur standard de JPEG2000 à l'endroit où l'interruption a été faite. Le fichier résultant peut alors être envoyé à travers tous types de canal de communication comme un autre fichier standard issu du codeur JPEG2000. Les différents niveaux de résolution et le caractère synchronisé de nos algorithmes permettent une visualisation en 3D, même avec peu de sous-bandes de résolution suite à un transfert partiel ou retardé. La méthode permet ainsi d'effectuer une visualisation à partir uniquement d'une fraction des données. Dans ce cas nous remplaçons par des zéros les coefficients des sous-bandes manquantes. La première phase de ce travail a concerné l'imperceptibilité; c'est la raison pour laquelle l'insertion a été réalisée dans les bits de poids plus faibles. La deuxième phase de ces travaux a concerné la robustesse, c'est pourquoi une stratégie d'insertion par étalement de spectres a été utilisée. Au cours de la seconde phase, l'imperceptibilité n'a pas été complètement ignorée, du fait que l'insertion des données est effaçable après l'extraction. Les deux applications principales sont la visualisation 3D de modèles numériques de terrains et de visages. Dans la troisième et dernière phase de ces travaux de thèse, nous avons élargi le problème en essayant de prendre en compte le problème d'assemblage de dalles de niveaux de résolutions différentes sans soudure apparente. Ceci a eté assuré par des fonctions de lissage dans le domaine des ondelettes.

**Mots clés :** 3D visualisation, hiérarchisation, synchronisation, adaptation, insertion des données cachées (IDC), LSB, étalement de spectre, multi-résolution, TOD, JPEG2000, sous-bande, MNT, 2.5D, texture, carte de profondeur, terrain, visage