

Visualisation 3D temps-réel à distance de MNT par insertion de données cachées basée ondelettes

K. Hayat¹

W. Puech^{1,2}

G. Gesquière³

M. Chaumont^{1,2}

¹ Laboratoire LIRMM, UMR CNRS 5506,
Université Montpellier II - France

² Centre Universitaire de Formation et de Recherche de Nîmes - France

³ Laboratoire LSIS, UMR CNRS 6168,
Université Aix-Marseille - France

khizar.hayat@lirmm.fr, william.puech@lirmm.fr,
gilles.gesquiere@lsis.org, marc.chaumont@lirmm.fr

Résumé

L'utilisation de photographies aériennes, d'images satellites, de cartes scannées et de modèles numériques de terrains amène à mettre en place des stratégies de stockage et de visualisation de ces données. Afin d'obtenir une visualisation en trois dimensions, il est nécessaire de lier ces images appelées textures avec la géométrie du terrain nommée Modèle Numérique de Terrain (MNT). Ces informations sont en pratiques stockées dans trois fichiers différents : MNT, texture, position et projection des données dans un système géo-référencé. Dans cet article, nous proposons de stocker toutes ces informations dans un seul fichier afin de les synchroniser. Nous avons développé pour cela une méthode d'insertion de données cachées basée ondelettes dans une image couleur. Les images de texture contenant les données MNT cachées peuvent ensuite être envoyées du serveur au client afin d'effectuer une visualisation 3D de terrains. Afin de combiner une visualisation en multirésolution et une compression, l'insertion des données cachées est intégrable dans le codeur JPEG 2000.

Mots clefs

Visualisation 3D, insertion de données cachées, compression JPEG2000, modèle numérique de terrains.

1 Introduction

Les Systèmes d'Information Géographique (SIG) sont des outils appréciés dans le domaine de l'aide à la décision dans de nombreuses entreprises et institutions. Ces SIG permettent en effet de combiner des données textuelles, des données vectorielles et des images (rasters). L'utilisation de données comme des photographies aériennes, des cartes scannées ou de modèles numériques de terrains implique de mettre en place des stratégies de stockage et de visualisation de ces données. En particulier, il devient difficile de stocker toutes ces données sur chaque ordinateur, surtout si

celui-ci est un média de faible capacité comme par exemple un pocket PC. De plus, ce problème de stockage est amplifié par l'évolution des capteurs qui permettent d'obtenir des images de meilleure qualité et donc de taille de plus en plus importante. Par exemple, il est actuellement possible de disposer d'images dont la précision au sol est inférieure au mètre. Cependant, ces données doivent pouvoir être transférées d'un serveur vers l'application utilisatrice. Pour le département des Bouches du Rhône par exemple, les photographies aériennes représentent plus de six Giga Octets d'information comprimée par JPEG 2000¹. De plus, la visualisation en trois dimensions d'un terrain implique de lier les images du sol, appelées textures, avec la géométrie du terrain, appelée modèle numérique de terrain (MNT). Ce lien est possible grâce aux coordonnées géo-référencées de ces éléments (longitude / latitude) qui dépendent du système de projection utilisé. Toutes ces informations sont en général stockées dans trois fichiers différents. L'objectif de nos travaux est de stocker et de synchroniser toutes ces informations dans un seul fichier afin de développer une application client-serveur de visualisation 3D temps-réel. A partir de cette approche il est possible de ne transmettre qu'un seul fichier regroupant toutes les informations et dépendant seulement de la zone où l'on se trouve. En fonction des débits de transmission et du point de vue de la visualisation 3D, un niveau de détail à transmettre est sélectionné. Afin de stocker toutes les informations dans un seul fichier, sans avoir à développer un nouveau format propriétaire, et de garder des performances d'un point de vue compression, nous proposons dans cet article d'utiliser des méthodes d'insertions de données cachées dans des images basées ondelettes. Ces travaux sont basés sur une version précédente d'insertion de données basée DCT [1]. Cet article est organisé de la manière suivante. Dans la sec-

¹L'Institut Géographique National (IGN) fournit les données raster au format JPEG 2000 ou au format Tiff

tion 2 nous introduisons les modèles numériques de terrains et leur visualisation 3D. Section 3, nous décrivons la décomposition en ondelettes d'une image et nous proposons une méthode d'insertion de données cachées. Enfin, section 4, nous présentons notre méthode sur des données réelles et analysons les résultats obtenus.

2 Représentation 3D des terrains

La représentation de terrains en trois dimensions est une composante importante pour la mise en place d'environnements graphiques extérieurs virtuels. Nous pouvons, par exemple, citer les simulateurs de vols et de conduite et de façon plus générale les applications multi-joueurs. Dans le cadre de nos travaux, nous nous intéressons à la visualisation 3D temps-réel de terrains. Ceci nous amène à combiner deux types de données [2]. Nous utilisons tout d'abord un champ d'altitudes. Chaque altitude correspond à une élévation (Figure 1.a). Celles-ci sont utilisées afin de générer la géométrie du terrain, en les connectant par exemple sous la forme de triangles (Figure 1.b). Nous devons ensuite plaquer une image sur ces triangles afin d'obtenir la visualisation voulue (Figure 1.c).

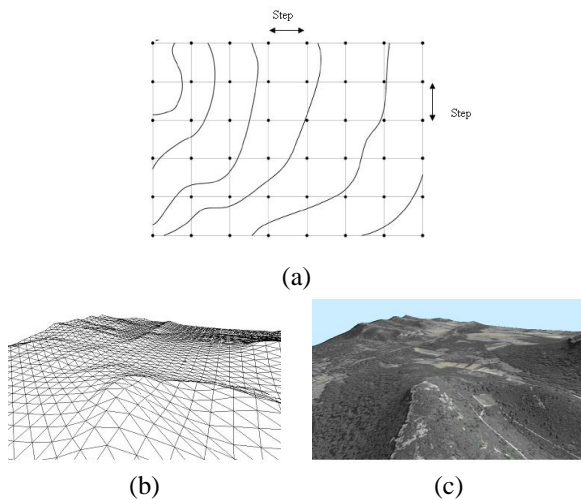


Figure 1 – a) Visualisation de la grille uniforme correspondante aux élévations de terrain, b) Surface 3D triangulée reliant les élévations, c) Plaquage d'une texture sur la géométrie.

2.1 Le modèle numérique de terrain (MNT)

Afin de représenter les altitudes correspondant à chaque point d'un terrain en trois dimensions, un champ d'élévation est utilisé. Celui-ci peut par exemple être représenté à l'aide d'une grille régulière dans laquelle on dispose d'une altitude tous les 50 mètres (Figure 1.a). Les points sont ensuite reliés afin de former des triangles que l'on pourra ensuite afficher. Le nombre de triangles à afficher est très important. Par exemple, plus de 10 millions de triangles sont nécessaires pour le département des Bouches du Rhône (environ 13000 Km^2).

De nombreuses méthodes ont été proposées afin de réduire le nombre de triangles fournis par une discrétisation uniforme, tout en préservant une bonne approximation de la surface initiale. Une première approche consiste à utiliser un ensemble de triangles irréguliers (TIN : Triangulated Irregular Network). Un grand nombre de méthodes a été développé afin de créer cet ensemble de triangles, comme par exemple la triangulation de Delaunay [3]. Des représentations hiérarchiques de ces triangulations ont été proposées afin d'obtenir le concept de niveau de détails [4]. Il est alors possible de diminuer le nombre de triangles tout en obtenant une précision identique. Une autre approche consiste à décomposer le terrain en un ensemble de grilles régulières de niveaux de détails différents [5].

2.2 Application de la texture

Une fois la construction géométrique effectuée, il faut plaquer sur les triangles obtenus grâce aux élévations des images des texture. La précision liée à de telles images est en général d'un pixel pour 50 cm sur le terrain, ce qui induit un coût important lié au stockage et au transfert des images vers le client. Il est donc nécessaire de penser à des stratégies de compression, de stockage et de visualisation rapide de ces données. De nombreuses méthodes de compression des images existent, les plus performantes, comme JPEG, induisent une détérioration de la qualité de l'image si le taux de compression est important. Plusieurs architectures client-serveurs ont été proposées afin de stocker un volume important de données utilisées sur plusieurs applications clientes [6]. Afin de visualiser rapidement ces données, une stratégie consiste à couper l'ensemble des images en une grille régulière [7] elle-même décomposée en niveaux de détails [8].

3 Insertion de données cachées basée ondelettes

Dans cet article, nous proposons de tirer partie de la compression en ondelettes de JPEG 2000. L'implémentation choisie sera basée sur la méthode de Lifting [9, 10]. Deux types d'ondelettes sont utilisées dans ce travail, une avec pertes (Daubechies 9/7) et une sans perte (Daubechies 5/3) [11]. Notre objectif est de visualiser, sur une application cliente, un terrain défini par des élévations ainsi que la texture correspondante. Ces données sont stockées sur un serveur distant. Les données sont envoyées en paquets de petites tailles afin de minimiser les temps d'attentes. Les données que nous utilisons sont en général stockées par l'IGN dans trois fichiers différents : le MNT, la texture et le système de coordonnées employé. Afin de stocker ces informations dans un fichier unique nous proposons d'insérer le MNT, ainsi que les coordonnées géo-référencées dans l'image de texture. Une information d'altitude est alors synchronisée avec un bloc de pixels de la texture. Nous proposons d'utiliser le format JPEG 2000 afin de profiter pleinement des propriétés de niveaux de détails sous-jacentes. De nombreuses méthodes ont été proposées

pour l'insertion de données cachées basée ondelettes, mais peu sont compatibles avec JPEG2000. Selon [12] pour l'insertion de données, les blocs de code doivent être traités séparément, c'est pourquoi des méthodes proposent des insertions inter-sous-bande [13] ou en multi-résolution hiérarchique [14]. D'autres approches proposent d'insérer les données cachées au niveau du flux binaire comprimé [15].

Afin d'insérer les informations d'altitude dans la carte de texture, nous proposons de suivre le protocole illustré figure 2. A partir de l'image de texture de N^2 pixels et de la carte de m^2 altitudes, nous en déduisons le facteur d'insertion $E = m^2/N^2$ coefficients/pixel. L'image de texture devra donc être découpée en bloc carré de $[1/E]$ pixels. Dans chacun de ces blocs, un coefficient d'altitude est donc caché. Dans un premier temps, la carte de texture est transformée en composantes Y, Cr, Cb. Une transformée en ondelettes discrètes (TOD) est appliquée sur les trois plans obtenus ainsi que sur la carte d'altitude. Nous choisissons d'utiliser une décomposition sans perte sur cette dernière afin de ne pas altérer les valeurs des altitudes.

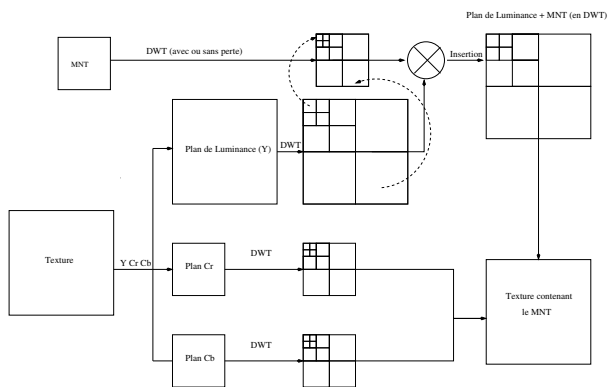


Figure 2 – Description de la méthode d'insertion des altitudes dans la carte de texture.

Afin d'assurer une cohérence spatiale entre les altitudes et la carte de texture, nous décomposons le plan de luminance Y, dans le domaine des ondelettes en bloc carré de $[1/E]$ coefficients quel que soit le niveau de décomposition. Dans chaque bloc nous insérons une information d'altitude du même niveau de décomposition. Nous retrouvons donc une mise en correspondance sur l'insertion dans les niveaux d'ondelettes. Par exemple, les basses résolutions de la carte d'altitude sont insérées dans les basses résolutions de la carte de texture. De ce fait, la transmission de la partie basse résolution de la carte de texture permet ainsi d'accéder directement à la partie basse résolution de la carte d'altitude. L'insertion des données se fait en modifiant les bits de poids faibles d'un certain nombre de coefficients du plan de luminance de la carte de texture. Ces coefficients sont choisis avec un générateur de nombres pseudo-aléatoire. A la réception le MNT est extrait de la carte de texture, même si seulement qu'une partie de l'image de texture a été transmise.

4 Résultats

Nous avons appliqué notre méthode sur une carte de texture des bouches du rhône de taille 2048×2048 pixels, illustrée figure 3.a, en y associant la carte d'altitude de 64×64 coefficients, illustrée figure 3.c. Un détail de 128×128 pixels de la carte de texture est représenté figure 3.b. Chaque coefficient d'altitude est codé sur 2 octets et le facteur d'insertion est donc de 1 coefficient pour 32×32 pixels de texture.

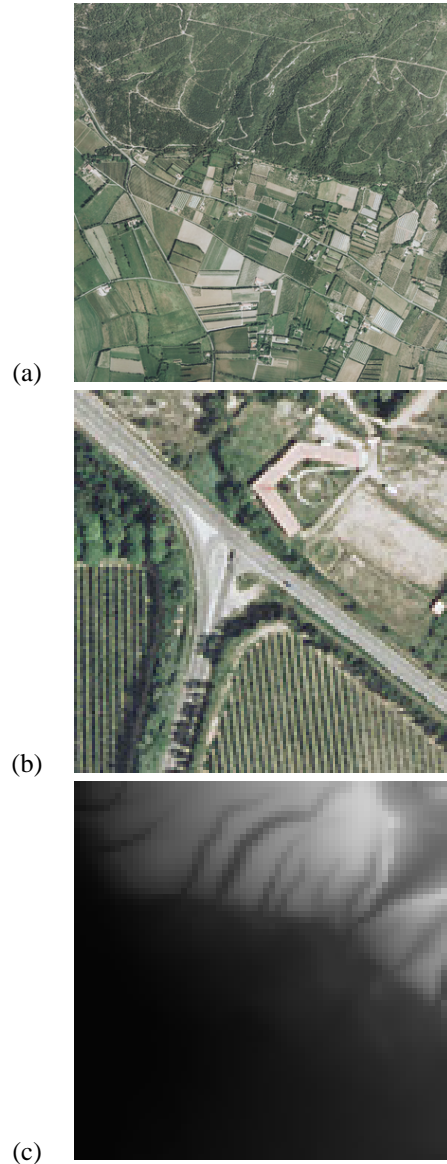


Figure 3 – Données originales : a) Texture, b) Détail texture, c) Altitudes.

En effectuant des décompositions en ondelettes avec pertes pour la carte de texture et sans perte par la carte d'altitudes nous obtenons les résultats illustrés figures 4 pour le niveau 1, et figures 5 pour le niveau 3 de décomposition. Pour la décomposition en ondelettes en un seul niveau, figure 4.a, l'insertion de données cachées se fera en quatre temps (LL, LH, HL et HH) alors que pour la décomposition

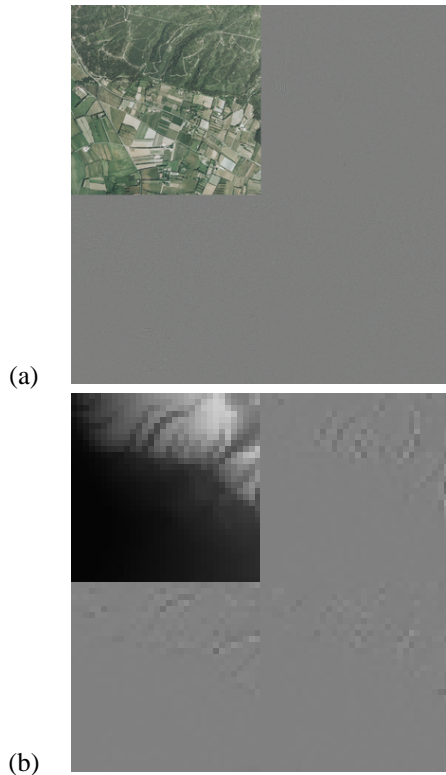


Figure 4 – *TOD, niveau 1 : a) Texture, b) Altitudes.*

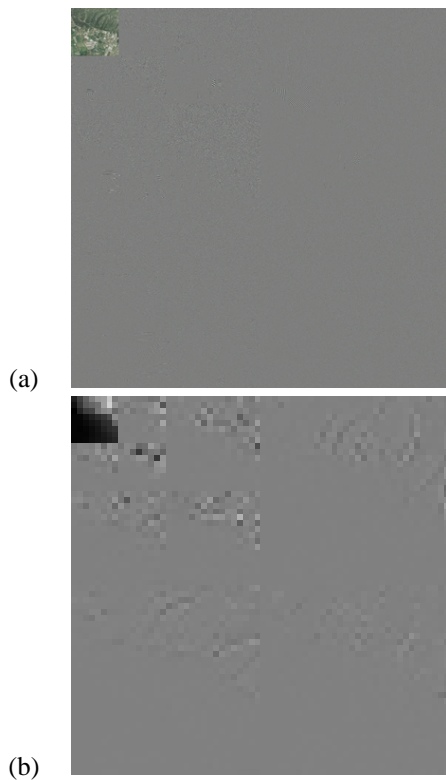


Figure 5 – *TOD, niveau 3 : a) Texture, b) Altitudes.*

en ondelettes en trois niveaux, figure 5.a, l'insertion de données cachées sera réalisée en dix étapes. Quel que soit le niveau de décomposition, la différence entre les images de luminance avant l'insertion des données et après l'insertion nous amène à un PSNR de 69.20 dB puisque nous insérons 16 bits d'un coefficient d'altitude dans 32×32 coefficients de texture sur la composante Y .

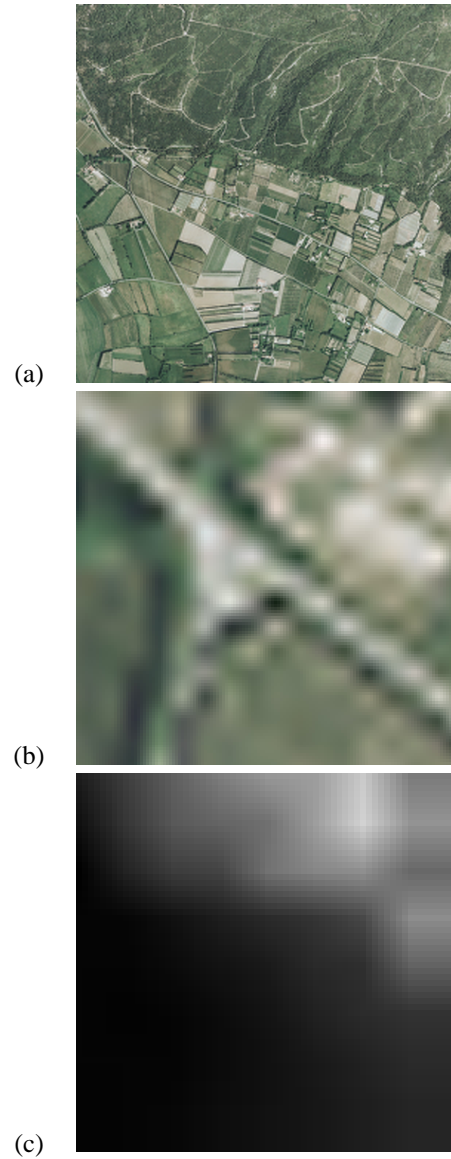


Figure 6 – *Reconstruction avec l'image d'approximation du niveau 3 : a) Texture, b) Détail texture, b) Altitudes extraites.*

A partir de l'image de textures décomposée jusqu'au troisième niveau et marquée avec les altitudes, si nous effectuons une TOD inverse uniquement avec l'image d'approximation du niveau 3, nous obtenons la carte de texture illustrée figure 6.a. En effectuant la différence entre cette carte de texture reconstruite et la carte de texture originale nous obtenons un PSNR de 20.90 dB. Un détail de la carte de texture reconstruite (identique à celui extrait figure 3.b)

est illustrée figure 6.b. A partir de l'image d'approximation du niveau 3, si nous effectuons l'extraction des données cachées suivie d'une TOD inverse, nous obtenons la carte d'altitudes illustrée figure 6.c. Le PSNR entre la carte d'altitude originale et celle reconstruite est alors de 29.25 dB. Notons qu'en utilisant que l'image d'approximation du niveau 3 nous n'avons eu besoin que de 1,6% des données initiales.

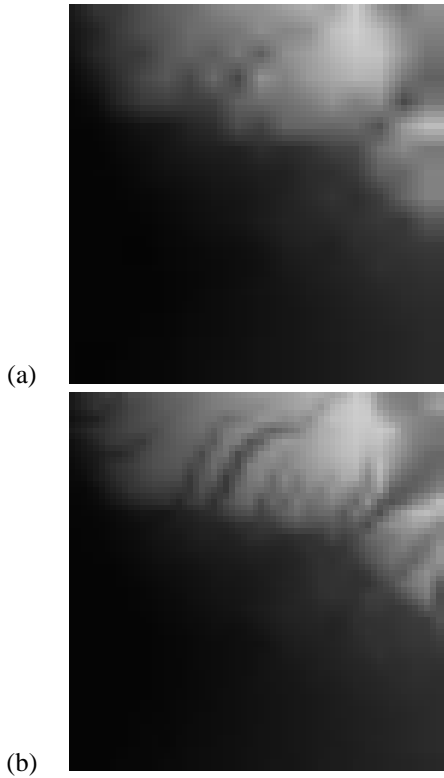


Figure 7 – Reconstruction de la carte d'altitudes : a) Avec le niveau 2, b) Avec le niveau 1.

	Niv. 3	Niv. 2	Niv. 1.	Tout
% données transmises	1.6 %	6.25 %	25 %	100 %
Texture (dB)	20.90	22.79	26.54	37.62
Altitude (dB)	29.25	33.51	40.37	∞
EQM Altitude (m^2)	77.37	29.00	5.97	0

Tableau 1 – Résultats obtenus pour l'extraction et la reconstruction en fonction de la quantité de données utilisée.

Le tableau 1 résume les résultats obtenus pour une reconstruction à partir des images d'approximation des niveaux 3, 2 et 1 ou en utilisant toutes les données. Notons que si nous utilisons toutes les données pour la TOD inverse nous obtenons un PSNR de 37.62 dB pour la carte de texture et un PSNR infini pour la carte d'altitude puisque nous avons utilisé une TOD sans perte. Les figures 7.a et b illustrent les cartes d'altitudes reconstruites respectivement avec les

images d'approximation des niveaux 2 et 1.

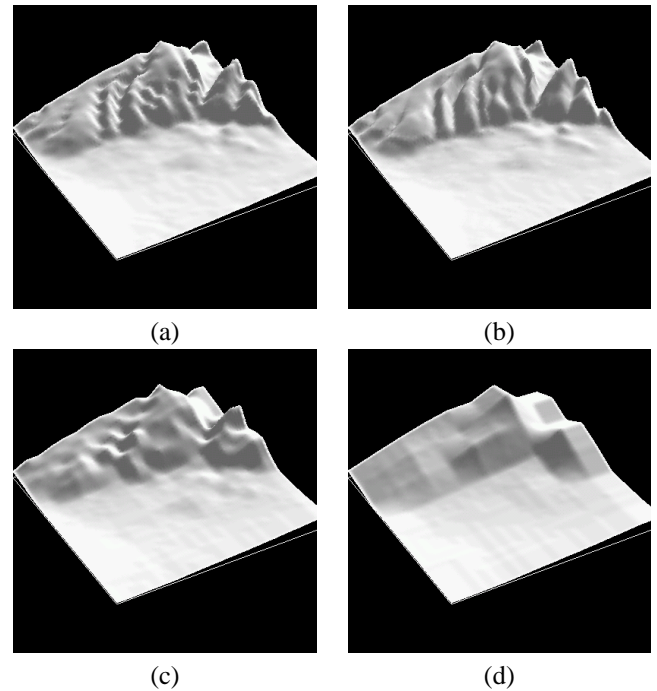


Figure 8 – Visualisation 3D des altitudes : a) Avec toutes les données, b) Avec le niveau 1, c) Avec le niveau 2, d) Avec le niveau 3.

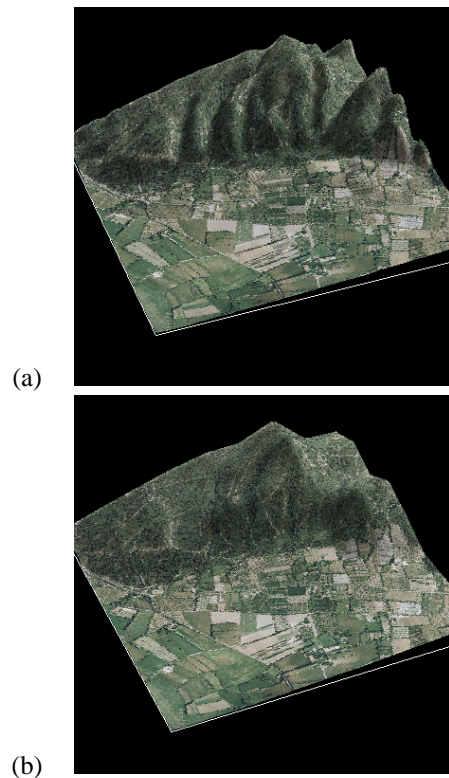


Figure 9 – Navigation 3D de la zone : a) Avec toutes les données, b) Avec le niveau 3.

Les figures 8.a, b, c et d représentent la reconstruction 3D des MNT, respectivement avec toutes les données initiales, qu'avec le niveau 1, qu'avec le niveau 2, et qu'avec le niveau 3. En plaquant la texture sur le MNT les figures 9.a et b permettent de comparer le résultat final entre une visualisation avec toutes les données et une visualisation uniquement avec le niveau 3 composé de 1.6 % des données initiales.

5 Conclusion

Dans cet article, nous avons présenté une nouvelle méthode d'insertion de données permettant de dissimuler un MNT dans l'orthophotographie qui lui est liée. Dans le cadre de l'application client-serveur que nous avons mise en place, nous pouvons ainsi synchroniser ces informations, évitant ainsi toute erreur liée à une combinaison erronée des données ou à un manque dû à leur transfert. De plus, l'utilisation d'images compressées sous la forme d'ondelettes permet de transférer le ou les niveaux nécessaires à une visualisation optimale en fonction du point de vue choisi par l'utilisateur, du débit et du média utilisé (ordinateur, pocket PC, ...) ou du niveau de détail choisi. De plus la méthode d'insertion des données développée est intégrable au codeur JPEG2000, de ce fait les images de texture peuvent être visualisées avec tous les lecteurs d'images JPEG 2000.

Bien que les résultats obtenus nous paraissent déjà intéressants, nous souhaiterions mettre en place, dans la suite de nos travaux, un stockage plus fin du modèle numérique de terrain. Nous pourrions par exemple nous intéresser aux ondelettes géométriques. Nous étudierions aussi la possibilité de ne plus utiliser des grilles uniformes à différents niveaux de détails, permettant ainsi de diminuer le nombre de triangle nécessaires à une bonne représentation du terrain si la variation de terrain est peu importante.

Références

- [1] A. Martin, G. Gesquiere, W. Puech, et S. Thon. Real Time 3D Visualisation of DEM Combined with a Robust DCT Based Data-Hiding Method. Dans *Electronic Imaging, Visualization and Data Analysis, SPIE, IS&T, San Jose, CA, USA*, volume 6060, pages 60600G–1–60600G–8, Jan 2006.
- [2] R. Raffin S. Thon, G. Gesquière. Visualisation 3D de feux de forêts sur des modèles numériques de terrain de l'IGN. Dans *Rencontre GEO-RISQUE 2006 "La cartographie des risques naturels"*, Montpellier (France), février 2006.
- [3] R. J. Fowler et J. J. Little. Automatic Extraction of Irregular Network Digital Terrain Models. Dans *SIGGRAPH '79 : Proceedings of the 6th annual conference on Computer graphics and interactive techniques*, pages 199–207, New York, NY, USA, 1979. ACM Press.
- [4] L. De Floriani et E. Puppo. Hierarchical Triangulation for Multiresolution Surface Description. *ACM Trans. Graph.*, 14(4) :363–411, 1995.
- [5] F. Losasso et H. Hoppe. Geometry Clipmaps : Terrain Rendering Using Nested Regular Grids. *ACM Trans. Graph.*, 23(3) :769–776, 2004.
- [6] P. Baumann, P. Furtado, R. Ritsch, et N. Widmann. Geo/Environmental and Medical Data Management in the RasDaMan System. Dans M. Jarke, M. J. Carey, K. R. Dittrich, F. H. Lochovsky, P. Loucopoulos, et M. A. Jeusfeld, éditeurs, *VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece*, pages 548–552. Morgan Kaufmann, 1997.
- [7] J. Yu et D. J. DeWitt. Processing Satellite Images on Tertiary Storage : A Study of the Impact of Tile Size on Performance. Dans *5th NASA Goddard Conference on Mass Storage Systems and Technologies*, 1996.
- [8] P. Hansen. OpenGL Texture-Mapping with Very Large Datasets and Multi-Resolution Tiles. Dans *SIGGRAPH '99 : ACM SIGGRAPH 99 Conference abstracts and applications*, page 262, New York, NY, USA, 1999. ACM Press.
- [9] W. Sweldens. The Lifting Scheme : a New Philosophy in Biorthogonal Wavelet Constructions. Dans *Electronic Imaging, Wavelet Applications in Signal and Image Processing, SPIE, IS&T, San Diego, CA, USA*, volume 2569, pages 68–79, Sep. 1995.
- [10] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.
- [11] I. Daubechies et W. Sweldens. Factoring Wavelet Transforms into Lifting Steps. *Fourier Anal. Appl.*, 4(3), 1998.
- [12] P. Meerwald et A. Uhl. A Survey of Wavelet-Domain Watermarking Algorithms. Dans *Electronic Imaging, Security and Watermarking of Multimedia Contents, SPIE, IS&T, San Jose, CA, USA*, Jan 2001.
- [13] D. Kundur. Improved Digital Watermarking Through Diversity and Attack Characterization. Dans *ACM Workshop on Multimedia Security'99, Orlando, FL, USA*, pages 53–58, Oct 1999.
- [14] D. Kundur et D. Hatzinakos. Digital Watermarking Using Multiresolution Wavelet Decomposition. Dans *International Conference on Acoustic, Speech and Signal Processing (IEEE ICASP 98), Seattle, Washington, USA*, May 1998.
- [15] P.-C. Su, H.-J. Wang, et 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*, 27(1-2) :35–53, June 1997.