

TECHNIQUES FOR PERSPECTIVE MAPPING OF PLANE TEXTURES
 COMPARAISON DE TECHNIQUES DE MISE EN PERSPECTIVE
 DE TEXTURES PLANES

Michel Gangnet et Djamchid Ghazanfarpour

Ecole Nationale Supérieure des Mines de Saint-Etienne, 158 Cours Fauriel 42023 Saint-Etienne Cedex.

RESUME

La présence de textures dans les images de synthèse de scènes tridimensionnelles est un élément important de réalisme. Si l'on considère la texture comme une image numérique, d'origine naturelle ou synthétique, plaquée sur un élément géométrique, face plane ou carreau bicubique, participant à la modélisation de la scène, des problèmes de changement de grille d'échantillonnage dus aux transformations géométriques et notamment à l'effet de perspective provoquent des phénomènes de repliement de spectre ou aliassage.

Dans cet article nous examinons certaines des méthodes proposées pour résoudre ces problèmes. Une typologie est décrite distinguant entre des méthodes de post-filtrage et des méthodes de filtrage a priori de la texture. Les méthodes de post-filtrage (CAT74, BLI76, CAT80, FEI80, GAN82) essaient de retrouver l'ensemble des informations relatives à une zone représentant un pixel de l'image finale, le post-filtrage étant une intégration numérique de ces informations. Les méthodes de filtrage a priori (NOR82, WIL83) supposent qu'il est possible de construire des versions de la texture permettant d'éviter les phénomènes d'aliassage par un filtrage passe-bas suivi d'un rééchantillonnage. Nous proposons diverses variantes de cette idée et montrons que les résultats obtenus, moins coûteux en temps de calcul, sont satisfaisants.

Le plan de l'article est le suivant:

1. Introduction
2. Post-filtrage
 - Nous discutons essentiellement les résultats de FEI80 et GAN82.
3. Filtrage a priori
 - 3.1 Taux de compression
 - 3.2 Filtrage global
 - 3.3 Filtrage adaptatif par découpage
 - 3.4 Représentation pyramidale (WIL83)
 - 3.5 Filtrage asymétrique
4. Conclusions

SUMMARY

Texture mapping is an important feature of 3D image synthesis. Texture is considered as a numerical picture mapped on elements of the scene such as planar faces or bicubic patches. In that case the mapping compounded with perspective transform will cause aliasing problems. The sampling grid of the final picture being constant, we must avoid high frequencies.

In this paper, we review some of the methods for texture anti-aliasing. We distinguish between post-filtering methods (CAT74, BLI76, CAT80, FEI80, GAN82) and a priori low-pass filtering (NOR82, WIL83).

We propose several methods of the last kind and show that the extension of WIL83 to non-symmetric filtering leads to good results.

Summary:

1. Introduction
2. Post-filtering
 - We mainly discuss the results of FEI80 and GAN82.
3. A priori filtering
 - 3.1 Compression rate
 - 3.2 Over-all filtering
 - 3.3 Adaptive filtering
 - 3.4 Pyramidal parametrics (WIL83)
 - 3.5 Non-symmetric filtering
4. Concluding remarks



1 - Introduction

Le problème de la représentation des textures dans les images de synthèse est une question déjà ancienne (CAT74). Il apparaît dès que l'on souhaite visualiser des scènes 3D contenant des surfaces auxquelles est affectée non pas une couleur uniforme mais une intensité lumineuse spatialement variable ou texture. Supposons que nous disposions d'une image représentant un mur recouvert de crépi vu de face en projection parallèle ; la figure 1 explicite les changements de repère nécessaires au calcul de l'intensité lumineuse en un point de l'image finale. Il en est de même pour l'application de textures sur d'autres surfaces comme des carreaux bicubiques.

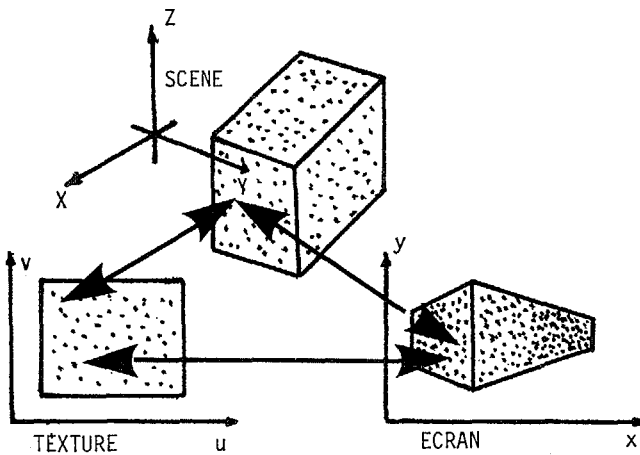


Figure 1 : mise en perspective d'une face plane portant une texture.

Fondamentalement, la mise en perspective d'une texture se ramène à un changement de variable dans le domaine spatial ; toutes les difficultés venant du fait que la version numérique de ce changement de variable est une modification de la grille d'échantillonnage et peut donc provoquer des phénomènes de repliement de spectre, ou aliassage, si les conditions d'application du théorème d'échantillonnage ne sont plus respectées.

Soit (x,y) les coordonnées de l'image finale et $I_a(x,y)$ la valeur de l'image analogique associée ; de même, (u,v) et $T_a(u,v)$ désignent les coordonnées de la texture et cette dernière sous sa forme analogique. La composition des applications de la figure 1 permet de définir l'application F de R^2 dans R^2 par $u = f(x,y)$, $v = g(x,y)$ et de poser :

$$I_a(x,y) = T_a(f(x,y), g(x,y)) \quad (1)$$

en tous les points (x,y) où F est définie.

Dans le cas, fréquent, où la texture est appliquée sur une face plane, F est une homographie de R^2 définie par :

$$u = \frac{ax+by+c}{mx+ny+p} ; \quad v = \frac{dx+ey+f}{mx+ny+p} \quad (2)$$

Le problème est donc de calculer les échantillons $I(i,j)$ de l'image finale aux points de coordonnées entières $x=i$, $y=j$ de sorte que l'image analogique $I^*_a(x,y)$ produite par le dispositif de reconstruction (ensemble convertisseur numérique-analogique, filtre de reconstruction, moniteur, oeil) soit suffisamment proche de $I_a(x,y)$.

La grille d'échantillonnage de l'image finale étant fixée, il est clair que I^*_a ne pourra contenir de fréquences supérieures à un seuil fixé, au moins théoriquement, par le théorème d'échantillonnage. Il est donc vain d'espérer reconstruire fidèlement I_a si celle-ci contient des fréquences non représentables. Nous pouvons par contre essayer de calculer l'image numérique I de sorte que I^*_a soit le résultat d'un filtrage passe-bas appliqué à I_a . Si l'on calcule $I(i,j)$ à l'aide de l'équivalent discret de (1) et si I_a contient des fréquences non représentables sur la grille de l'image finale, l'existence de basses fréquences parasites dans le signal reconstitué I^*_a produira des effets d'aliassage.

L'importance des effets d'aliassage dans l'image échantillonnée puis reconstruite est liée au pourcentage de l'énergie totale de l'image initiale située dans la partie non représentable du spectre. Par ailleurs, le système oeil-cerveau est davantage sensible aux défauts présents dans les images structurées. Ces deux phénomènes sont visibles sur les figures 2, 3, 4 et 5. L'image 2 est le résultat d'une synthèse et présente de nombreux changements abrupts d'intensité. L'image 3 est le résultat d'une numérisation par caméra, l'effet de cette opération étant un filtrage passe-bas de l'image initiale. Les images 4 et 5 sont obtenues en prenant pour valeur d'un pixel (i,j) celle du plus proche voisin de son image par F , identique dans les deux cas. On observe que les effets d'aliassage, déstructurant complètement 4, sont beaucoup moins importants sur 5.

Revenons à l'objectif fixé plus haut : construire une image numérique I de sorte que I^*_a soit le résultat d'un filtrage passe-bas appliqué à I_a . Deux approches sont possibles : le post-filtrage d'une image numérique suréchantillonnée ou un filtrage a priori de la texture permettant une utilisation directe de (1).

Post-filtrage : I_a contenant des hautes fréquences non représentables, on détermine les pas d'échantillonnage dx et dy compatibles avec le spectre de I_a . On échantillonne I_a sur la grille de calcul de pas dx, dy avec (1) ; soit C le résultat. Comme on ne peut reconstruire une image analogique à partir de C , on calcule l'image numérique I à la résolution de la grille d'affichage à partir de C par un filtrage numérique passe-bas. Les méthodes décrites dans la littérature sous le nom de "filtering before sampling" sont de ce type, en effet c'est le rééchantillonnage de l'image finale qui est visé.

Filtrage a priori : le pas d'échantillonnage de la grille d'affichage est fixé. Au lieu d'échantillonner T_a on échantillonne une image T^*_a qui est le résultat d'un filtrage passe-bas appliqué à T_a . Ce filtrage passe-bas doit être compatible avec l'effet sur le spectre de T_a de la transformation géométrique.

Dans les deux cas, pour éviter le recours à des calculs dans le domaine de Fourier, la détermination des pas dx, dy ou de la fréquence de coupure du filtre passe-bas est effectuée en considérant un taux de compression lié à la transformation géométrique.

2 - Post-filtrage

FEI80 définit tout d'abord un masque de convolution centré sur chaque pixel de l'image finale. Supposons pour simplifier que le masque choisi ait une base rectangulaire B et que F soit donné par (2). Les auteurs proposent de calculer $F(B)$, qui est dans ce cas un quadrilatère, puis de sélectionner les pixels intérieurs à $F(B)$ dans l'image de la texture. Après avoir calculé l'image par F^{-1} de ces pixels, le résultat est convolué numériquement avec le masque. Cette dernière opération est réalisée à l'aide d'une table donnant les poids du masque pour un suréchantillonnage régulier de B . Les poids affectés aux valeurs ne figurant pas dans la table sont déterminés par celui du plus proche voisin.

GAN82 utilise une méthode légèrement différente mais se ramenant aussi à un suréchantillonnage local de l'image finale. B est maintenant un carré de côté 1 centré en (i, j) . La longueur de la plus grande diagonale de $F(B)$ est utilisée pour déterminer le pas de suréchantillonnage de B . L'intensité en un point de la sur-grille régulière est obtenu par une approximation bilinéaire dans l'image de la texture. L'ensemble des intensités obtenues est ensuite filtré par un masque en sinc bidimensionnel pour obtenir l'intensité en un point de la grille d'affichage. La figure 6 montre un résultat obtenu par cette méthode, la texture étant celle de la figure 2. On remarquera la persistance d'effets d'aliassage dans les zones très comprimées due à un échantillonnage insuffisant. Ces deux méthodes sont coûteuses en temps de calcul puisqu'il est nécessaire de calculer les images par F ou F^{-1} de très nombreux points. Il est possible d'optimiser le calcul de (2) par un calcul incrémental ne demandant que des additions et deux divisions par point. La première méthode somme toute l'information disponible relative à un pixel de l'image finale mais utilise une convolution sur une grille irrégulière alors que la deuxième méthode interpole l'information mais somme sur une grille régulière. Dans les deux cas des effets d'aliassage peuvent persister; d'autre part le volume de calcul est plus important dans la première méthode. BLI76 et CAT80 décrivent d'autres méthodes. Le filtre utilisé dans BLI76 est une pyramide à base carrée B . En approximant localement F par son application linéaire tangente, les auteurs calculent l'intensité des points de texture

contribuant à B . Ces intensités sont pondérées par les altitudes de la pyramide déformée par F . Le filtrage est donc réalisé sur la texture. CAT80 explicite une décomposition de la transformation F^{-1} en deux passes indépendantes sur x et y , la méthode est très intéressante du point de vue du temps de calcul mais nous n'avons pas testé son comportement pour des compressions élevées.

3 - Filtrage a priori

Pour diminuer les temps de calcul des méthodes précédentes, on peut penser à utiliser des versions basses fréquences de l'image de la texture. Cette technique a été proposée par L. Williams dans WIL83, dans la suite nous examinons diverses variantes de cette idée. De façon informelle, on peut dire qu'un échantillonnage par point utilisant (1) produira moins d'aliassage s'il est appliqué à un exemplaire de T_a ayant subi un filtrage passe-bas. Le problème est de déterminer le filtre à utiliser ; une fréquence de coupure trop basse provoquera du flou, une fréquence de coupure trop élevée des phénomènes d'aliassage. Le filtrage a priori de la texture doit donc être compatible avec la transformation géométrique.

3.1 Taux de compression

Considérons une transformation perspective d'un signal monodimensionnel, fig. 7, et supposons que le signal $T_a(u)$ puisse être restitué fidèlement à partir des échantillons choisis. A gauche du point A on a $F(dx) < du$, dans cette zone le signal $T_a(F(x))$ est suréchantillonné, il suffira de calculer les échantillons de $T_a(F(x))$ par une approximation linéaire. A droite de A , $F(dx) > du$, le signal $T_a(F(x))$ est souséchantillonné. Dans le cas monodimensionnel, le filtrage passe-bas effectué sur T_a doit donc être localement proportionnel au rapport $F(dx)/du$.

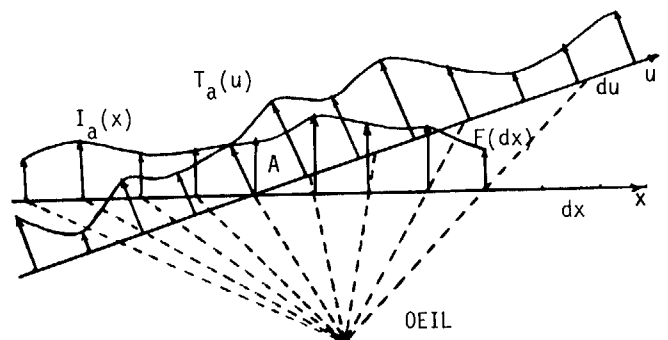


Figure 7 : perspective d'un signal mono-dimensionnel

L'argument précédent peut être étendu au cas bidimensionnel. F étant une application de R^2 dans R^2 suffisamment régulière, l'image par F , application linéaire tangente à F , d'un carré de côté 1 centré en (x, y) est un parallélogramme. Les longueurs des côtés de ce parallélogramme, projetées sur les axes (u, v) sont : sur l'axe u , $|f'_x|$ et $|f'_y|$, sur l'axe v ,

$$|g'_x| \text{ et } |g'_y|.$$



M. Gangnet et D. Ghazanfarpour

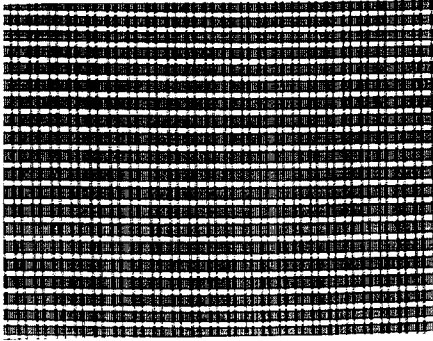


Figure 2 : texture synthétique haute fréquence.



Figure 3 : texture basse-fréquence numérisée par caméra.

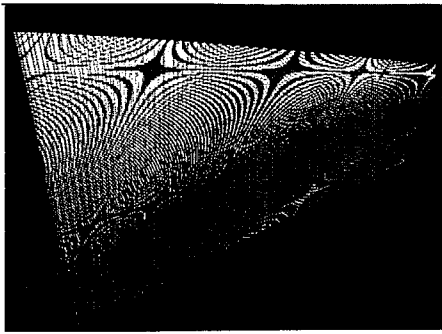


Figure 4 : mise en perspective sans traitement d'antialiasage.

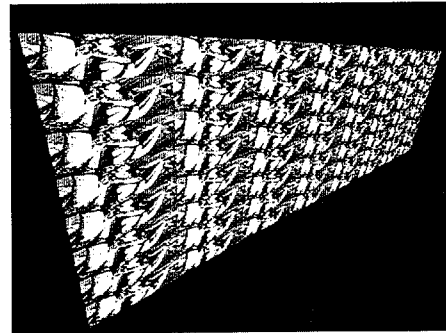


Figure 5 : mise en perspective sans traitement d'antialiasage.

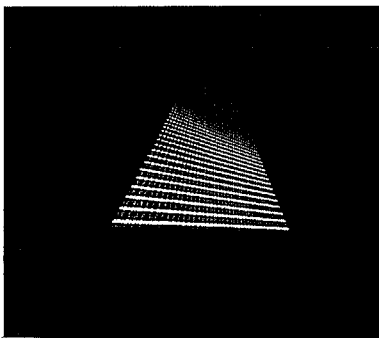


Figure 6 : mise en perspective avec post-filtrage.

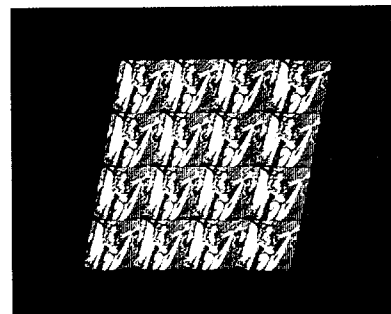


Figure 8 : mise en perspective avec filtrage global.

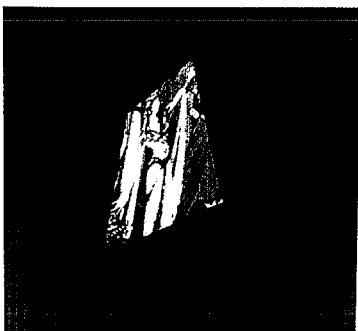


Figure 10 : mise en perspective avec filtrage adaptatif par découpage.



Figure 11 : zoom x4 de la figure 10.

Nous définissons les taux de compression en (x,y) par $D_U = \text{Sup}(|f'_x|, |f'_y|)$, $D_V = \text{Sup}(|g'_x|, |g'_y|)$ et $D = \text{Sup}(D_U, D_V)$. Un choix légèrement différent de D est proposé dans WIL83. Si nous utilisons un filtre passe-bas symétrique dont l'action est proportionnelle à D , la texture sera filtrée symétriquement. Au risque de produire un effet de flou dans celle des deux directions qui est la moins comprimée. Il est donc préférable d'utiliser un filtre passe-bas asymétrique.

3.2 Filtrage global

Si la compression de la texture due à la transformation perspective est considérée comme constante en tous les points de l'image finale, on obtient :

$$I(i,j) = T'(f(i,j), g(i,j)) \quad (3)$$

où T' est la texture filtrée. Les coefficients et la taille du filtre sont obtenus à partir des taux de compression. La valeur de T' au point image de i,j par F étant obtenue par approximation bilinéaire. Le résultat d'un tel traitement est présenté sur la figure 8. Cette méthode très simple peut cependant être appliquée avec succès à des facettes de taille raisonnable dont l'angle avec le plan de l'écran est faible. Dans le cas de (2), il suffit de calculer le taux de compression aux sommets du rectangle englobant de la face à texturer dans l'image finale.

3.3 Filtrage adaptatif par découpage

Dans le cas où F est la transformation définie par (2), on peut tirer profit de ses propriétés projectives. Soit P un polygone à texturer dans l'image finale, connaissant les images U et V par F^{-1} des points à l'infini des axes u et v de la texture, il est toujours possible de trouver un quadrilatère $ABCD$ contenant P tel que $ABCDUV$ soit un quadrilatère complet du plan projectif.

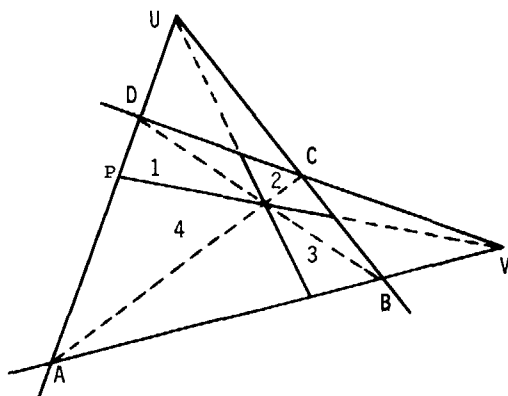


Figure 9 : division d'un quadrilatère.

F conserve le birapport de 4 points alignés, ADPU est donc une division harmonique; ceci permet de découper $ABCD$ en 4 quadrilatères. Ce procédé est récursif. Le test d'arrêt peut être défini par les dimensions des fils et la proximité de leur taux de compression avec celui de leur père, le taux de compression d'un quadrilatère étant pris égal à celui de son centre.

L'image par F de chacun des quadrilatères est un rectangle dans l'image de la texture, ce qui facilite le filtrage passe-bas. Chaque quadrilatère est alors rempli par une approximation bilinéaire effectuée sur le rectangle correspondant filtré. On ne traite évidemment que les pixels intérieurs à P . Un résultat de la méthode est présenté figure 10. La figure 11 qui est une $x4$ de 10 montre que la structure de l'image de départ n'a pas été perdue. On voit que les différents filtrages dépendent de F et devront donc être répétés pour chaque nouvelle transformation. Les méthodes suivantes évitent cet inconvénient.

3.4 Représentation pyramidale des textures

La méthode exposée par L. Williams dans WIL 83 propose de générer pour une texture donnée de dimensions $2^n \times 2^n$ une suite d'images T_p de dimensions $2^p \times 2^p$, $0 \leq p < n$. Chaque pixel de T_{p-1} étant la moyenne simple des $2 \times 2 = 4$ pixels correspondant de T_p . La fig.12 montre la suite T_p , $0 \leq p < 8$, la texture de base étant celle de la figure 2. Sur la figure 13, $x5$ de 12, on voit que la fenêtre de Fourier utilisée ne supprime pas suffisamment de hautes fréquences pour que la texture soit bien représentée sur toutes les images T_p . Il est vrai que la texture de base est particulièrement difficile à représenter. Les images T_p correspondent à des taux de compression de 2^{n-p} , égaux en u et v . La méthode utilisée pour le calcul du pixel (i,j) est la suivante:

- calculer le taux de compression D en $F(i,j)$, unique pour les deux directions ;
- déterminer p tel que $2^{n-p-1} < D < 2^{n-p}$;
- calculer les intensités de $F(i,j)$ sur T_p et T_{p-1} par deux approximations bilinéaires et la valeur finale par une approximation linéaire.

L'intérêt évident de cette méthode est que la texture est filtrée une fois pour toute et que la représentation pyramidale peut être utilisée quelle que soit la transformation géométrique. L. Williams remarque que l'utilisation d'une fenêtre de Fourier pour la construction de la structure pyramidale est une utilisation minimale de la théorie du signal et note qu'un filtrage symétrique peut être source de problèmes. Ceci est confirmé par les améliorations qu'apporte l'utilisation de deux taux de compression différents en u et v .

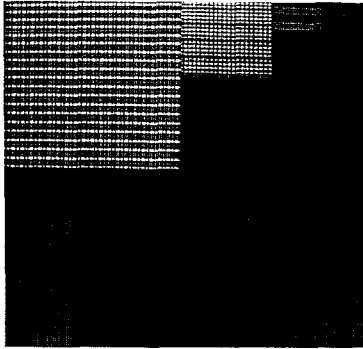


Figure 12 : représentation des textures avec filtrage symétrique.

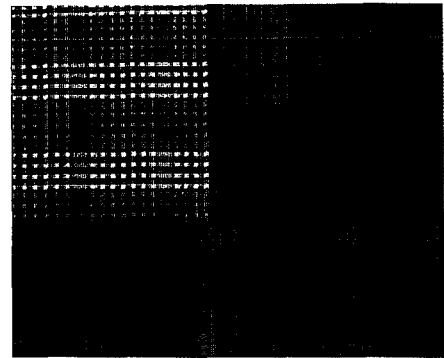


Figure 13 : zoom x5 de la figure 12.

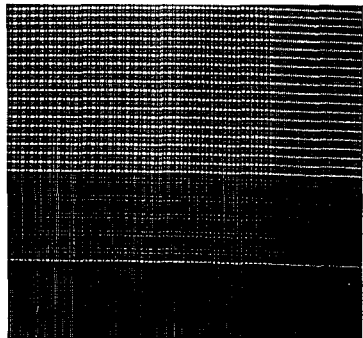


Figure 14 : représentation des textures avec filtrage asymétrique.

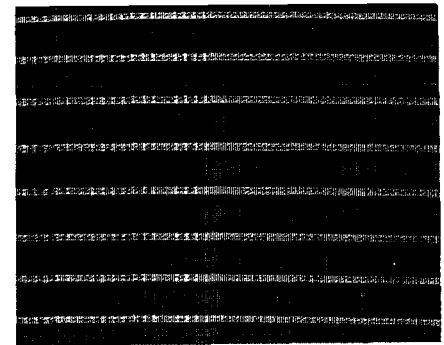


Figure 15 : zoom x5 de la figure 14.

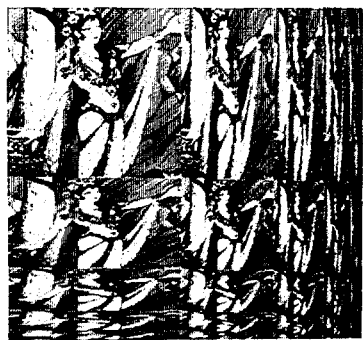


Figure 16 : représentation des textures avec filtrage asymétrique.

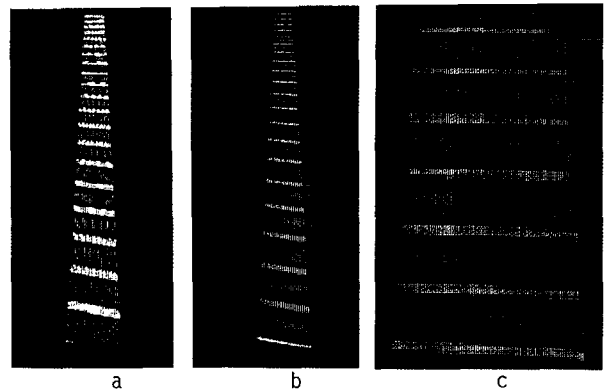


Figure 17 : mise en perspective, (a) sans traitement d'antialiasage, (b) avec filtrage asymétrique, (c) zoom x7 de (b).



Figure 18 : mise en perspective avec filtrage asymétrique.



Figure 19 : zoom x4 de la figure 18.

3.5 Filtrage asymétrique des textures

Soit $T_{n,n}$ la texture de base de dimension $2^n \times 2^n$, nous proposons de construire la suite d'images $T_{p,q}$ de dimension $2^p \times 2^q$, $0 < p < n$ et $0 < q < n$. L'image $T_{p,q}$ étant obtenue soit par sous-échantillonnage de l'image $T_{p+1,q}$, filtrée suivant u soit par sous-échantillonnage de l'image $T_{p,q+1}$ filtrée suivant v . Le filtrage appliqué est un filtrage passe-bas monodimensionnel utilisant une fonction de pondération. Algorithmiquement ces images peuvent être générées en deux passes : une passe horizontale de construction des $T_{p,n}$, $0 < p < n$ suivie d'une passe verticale qui, pour chaque $T_{p,n}$ calculé précédemment, génère la suite $T_{p,q}$, $0 < q < n$. Les figures 14 et 16 présentent les images $T_{p,q}$ $0 < p < 8$ et $0 < q < 8$ pour les textures de base des figures 2 et 3. La figure 15 est un zoom $\times 5$ de la figure 14. Une telle structure va permettre de calculer les taux de compression indépendamment dans les deux directions. La méthode utilisée pour le calcul du pixel (i,j) est la suivante :

- calculer les taux de compression D_u et D_v
- déterminer p et q tels que $2^{n-p-1} < D_u < 2^{n-p}$ et $2^{n-q-1} < D_v < 2^{n-q}$.
- calculer les valeurs de $F(i,j)$ sur $T_{p,q}$, $T_{p-1,q}$, $T_{p,q-1}$ et $T_{p-1,q-1}$ par quatre approximations bilinéaires et la valeur finale par une approximation bilinéaire.

La figure 17 présente les résultats, sur la texture de base de la figure 2, contenant beaucoup d'énergie dans les hautes fréquences, pour une transformation F qui induit un taux de compression très différent pour chacune des deux directions. L'image 17.a visualise l'application de F sans traitement, c'est à dire en affectant au pixel (i,j) la valeur du plus proche voisin de son image par F sur la texture de base. Les figures 17.b et 17.c présentent les résultats de la méthode précédemment décrite pour la même transformation, 17.c étant un zoom $\times 7$ de 17.b. L'application sur une texture basse fréquence pour une transformation F dont les taux de compression sont moins importants est visualisée sur les figures 18 et 19, 19 étant un zoom $\times 4$ de 18.

Les approximations linéaires ou bilinéaires utilisées dans les diverses méthodes gagnent à être réalisées par (4).

$$(1-b)((1-a)p+aq) + b((1-a)r+as) = p+a(q-p)+b(r-p)+ab(s-r-q+p) \quad (4)$$

En virgule fixe, (4) peut être obtenue par des lectures dans des tables précalculées.

Signalons enfin que NOR82 étudie le cas où la texture est connue par son développement en série de Fourier. Les auteurs montrent que l'on peut calculer l'effet de F sur les signaux élémentaires et proposent de remplacer la texture en perspective par une valeur constante à partir du moment où elle n'est plus représentable sur la grille d'affichage.

4 - Conclusions

Nous avons examiné deux types de méthodes pour résoudre le problème de changement de grille

d'échantillonnage posé par des transformations géométriques appliquées à des images numériques. La première classe de méthodes est en fait une série de variations sur le calcul numérique de (5) où h est une fonction de pondération.

$$I(i,j) = \iint h(i-x,j-y)T_a(f(x,y),g(x,y)) dx dy \quad (5)$$

Pour un pixel, les calculs nécessaires sont en gros proportionnels au taux de compression local. La deuxième classe de méthodes fait l'hypothèse que des versions de la texture adaptées au taux de compression local sont disponibles et construit l'image finale par des approximations. Le gain en temps de calcul est évident puisque la représentation pyramidale de la texture étant construite une fois pour toute, le nombre d'opérations par pixel est constant.

Les résultats obtenus par filtrage a priori étant convenables, il apparaît qu'à condition d'utiliser un préfiltrage asymétrique, nous disposons d'une méthode fiable, susceptible d'implémentation microprogrammée dans le cas où les textures sont définies par des vignettes. Si les textures utilisées sont suffisamment filtrées ou comportent peu d'énergie dans les hautes fréquences, les approximations bilinéaires pourront être remplacées par des arrondis sur le plus proche voisin. C'est souvent le cas pour des textures représentant des éléments naturels: végétation, nuages. Dans notre environnement visuel, l'homme est généralement responsable des hautes fréquences spatiales, la nature est heureusement plus clémente.

5 - Remerciements

Nous remercions Sabine Coquillard pour des discussions utiles et son aide dans la mise au point de la méthode de filtrage adaptatif par découpage.

6 - Bibliographie

- BLI76 J. BLINN and M. NEWELL,
"Texture and Reflection in Computer Generated Images", Comm. ACM, 19-10, Oct.1976.
- CAT74 E. CATMULL,
A Subdivision Algorithm for Computer Display of Curved Surfaces, PhD thesis, University of Utah, Dec.1974.
- CAT80 E. CATMULL and A. SMITH,
"3-D Transformation of Images in Scanline Order", Comput. Graph., 14-3, Jul.1980.
- FEI80 E. FEIBUSH, M. LEVOY and R. COOK,
"Synthetic Texturing Using Digital Filters", Comput. Graph., 14-3, Jul.1980.
- GAN82 M. GANGNET, D. PERNY and Ph. COUEIGNOUX,
"Perspective Mapping of Planar Textures", Eurographics 1982, North-Holland, 1982.
- NOR82 A. NORTON, A. ROCKWOOD and P. SKOLMOSKI,
"Clamping: A Method of Antialiasing Textured Surfaces by Bandwidth Limiting in Object Space", Comput. Graph., 16-3, Jul.1983.
- WIL83 L. WILLIAMS,
"Pyramidal Parametrics", Comput. Graph., 17-3, Jul.1983.