



**HAL**  
open science

## Agrandissement d'image basé sur une carte directionnelle multirésolution

Eric van Reeth, Pascal Bertolino, Marina Nicolas

► **To cite this version:**

Eric van Reeth, Pascal Bertolino, Marina Nicolas. Agrandissement d'image basé sur une carte directionnelle multirésolution. GRETSI 2011 - XXIIIème Colloque francophone de traitement du signal et des images, Sep 2011, Bordeaux, France. pp.CD. hal-00593233v2

**HAL Id: hal-00593233**

**<https://hal.science/hal-00593233v2>**

Submitted on 17 Jun 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Agrandissement d'image basé sur une carte directionnelle multirésolution

Eric VAN REETH<sup>1,2</sup>, Pascal BERTOLINO<sup>1</sup>, Marina NICOLAS<sup>2</sup>

<sup>1</sup>GIPSA-lab - Grenoble INP - France

<sup>2</sup>STMicroelectronics, 12 rue Jules Horowitz, Grenoble, FRANCE

eric.vanreeth@gmail.com,  
pascal.bertolino@gipsa-lab.grenoble-inp.fr,  
marina.nicolas@st.com

**Résumé** – Nous proposons un schéma novateur d'interpolation d'images, basé sur une analyse multirésolution de la direction des contours. Le but est de corriger les artefacts classiques des méthodes habituelles (bilinéaire, bicubique), tout en évitant l'apparition des artefacts engendrés par la plupart des méthodes d'interpolation directionnelle. Notre estimation d'orientation de contours est basée sur une division de l'image originale en *quadtree* et une étude fréquentielle des contours. L'interpolation en elle-même est basée sur l'utilisation d'une interpolation isotrope (spline cubique), corrigée grâce à un filtrage Gaussien localement orienté dans la direction des contours. Les régions sans contours sont préservées grâce à la création d'un masque construit à partir de filtres de Gabor. Enfin, les résultats obtenus sont comparés à des méthodes d'interpolation directionnelle récentes, afin d'illustrer les bonnes performances de notre algorithme sur des images naturelles de natures variées.

**Abstract** – We propose a new image interpolation process, based on a multiresolution edge orientation analysis. The goal of this technique is to correct usual artifacts yielded by classical methods (bilinear, bicubic), without introducing new artifacts that are often produced by directional interpolations. Our orientation estimation is based on a quadtree division and a multiresolution approach. Our interpolation technique is based on an isotropic reference interpolation (cubic-spline) that is corrected by a Gaussian filter locally oriented in the direction of the edge. Edge-free regions are preserved with a Gabor mask that is built to protect pixels which do not need any correction. Finally, our results are compared to recent state-of-the-art directional interpolations to illustrate the good performance of our algorithm on various contents of natural images.

## 1 Introduction

L'agrandissement d'image est largement utilisé depuis une vingtaine d'années : zoom numérique, affichage d'images de définition inférieure à la définition de l'écran, ... Les techniques d'interpolation peuvent être divisées en deux catégories : adaptatives ou non. Les méthodes non adaptatives (bilinéaire, bicubique, spline, ...) sont généralement rapides et faciles à implémenter mais ne parviennent pas à donner des résultats satisfaisants pour tous types de contenu. Des artefacts bien connus comme le *jagging*, le crènelage et le flou apparaissent alors en particulier au niveau des contours. Les méthodes adaptatives essaient de limiter l'apparition de ces artefacts en adaptant leur traitement en fonction des propriétés des pixels.

La méthode adaptative présentée vise à ajuster l'interpolation en fonction de la direction du contour sur lequel se trouve le pixel à traiter. La direction des contours est en effet une information très utile puisqu'elle indique la direction le long de laquelle les variations des pixels sont douces (direction parallèle au contour), et à l'inverse la direction le long de laquelle les variations sont franches (direction perpendiculaire au contour). Adapter l'interpolation en fonction de ces caractéristiques permet de conserver au mieux l'aspect des contours de l'image

basse-définition dans une grille plus définie. Nous présentons dans un premier temps notre technique de recherche de la direction des contours, puis dans un deuxième temps notre méthode d'interpolation basée sur les directions détectées.

## 2 Calcul de la carte directionnelle

Notre approche s'inspire du travail de Peyré et al [1] pour la construction des bases de bandelettes. Dans cette méthode, l'image est partitionnée en blocs de taille variable selon un algorithme de *quad-tree*. L'objectif est d'isoler au plus une direction dans chaque bloc, pour définir l'orientation avec laquelle les bandelettes sont calculées. L'algorithme proposé par Peyré consiste à minimiser un critère de variation de manière itérative pour obtenir la partition optimale. Ce critère est d'abord calculé dans les plus petits blocs qui peuvent fusionner si leur regroupement entraîne la création d'un bloc parent qui fait diminuer la valeur du critère.

À l'inverse, notre méthode estime d'abord les variations dans les grands blocs. Ces derniers sont divisés uniquement dans le cas où ils contiennent plus d'une direction de contour. Cette approche présente trois avantages principaux :

- les variations des blocs de taille inférieure ne sont calculées

lées que si nécessaire,

- le calcul de variation est plus robuste lorsqu’il est effectué dans un grand bloc,
- la précision sur l’angle estimé est meilleure car la résolution angulaire est plus élevée dans les grands blocs.

Les parties suivantes décrivent comment dans chaque bloc, la direction du contour est calculée. La première étape consiste à adapter la résolution du bloc à celle du contour. La deuxième explique comment le bloc est projeté sur des segments de droite 1D afin d’étudier les variations dans les différentes directions.

## 2.1 Adaptation de résolution

Le but de cette étude est d’adapter localement la résolution d’un bloc en fonction des caractéristiques fréquentielles du ou des contours contenus dans le bloc. Pour cela, une transformée en ondelettes isotrope non décimée (IUWT) est utilisée [2]. Les propriétés isotropes de cette transformée sont nécessaires pour ne favoriser aucune direction lors de cette étape préalable à la détection de direction en elle-même. Le fait d’utiliser une transformée non-décimée permet de conserver des tailles de blocs similaires, et donc une résolution angulaire identique à travers les échelles. Notons enfin que seuls trois niveaux de résolutions sont utilisés. Nous considérons en effet que la plupart des contours présents dans les images naturelles peuvent être représentés de manière efficace dans les images de détails à l’une des trois premières échelles. Par la suite, l’image est découpée en blocs réguliers de taille  $(16 \times 16)$  pixels et l’échelle la mieux adaptée est choisie pour chaque bloc. L’échelle optimale,  $J_{opt}$  est celle qui maximise la moyenne  $M_j$  de l’amplitude des coefficients d’ondelettes du bloc. C’est l’échelle pour laquelle la corrélation entre la fréquence de l’ondelette et la fréquence des contours est la plus élevée. Soient  $M_j^v$  et  $M_j^h$  respectivement les moyennes des amplitudes de chaque colonne et de chaque ligne du bloc  $\Delta_j$  de taille  $N \times N$  à l’échelle  $j$ .

$$M_j^v = \frac{\sum_{x=1}^N (\sup_{y=1\dots N}(\Delta_j(x, y)) - \inf_{y=1\dots N}(\Delta_j(x, y)))}{N}$$

$$M_j^h = \frac{\sum_{y=1}^N (\sup_{x=1\dots N}(\Delta_j(x, y)) - \inf_{x=1\dots N}(\Delta_j(x, y)))}{N}$$

$$M_j = \frac{M_j^v + M_j^h}{2}$$

$$J_{opt} = \operatorname{argmax}_{j=1\dots 3}[M_j]$$

Les figures 1(a)-(c) représentent les images de détails de la transformée IUWT d’une portion d’une image naturelle dont la résolution décroît de gauche à droite. La figure 1(d) illustre l’image composée pour chaque bloc d’une des trois résolutions grâce au critère défini plus haut. Remarquons que les zones de contours hautes fréquences (foulard) sont représentées par la résolution la plus fine ( $J_{opt} = 1$ ), alors que les zones de contours basses fréquences (visage) sont représentées par la résolution la plus basse ( $J_{opt} = 3$ ). Notons enfin que si l’image composite présente des frontières de blocs évidentes, ceci n’est

pas gênant par la suite car l’étude directionnelle est effectuée indépendamment dans chacun des blocs.



FIGURE 1 – (a)-(c) Images de détails de résolution décroissante. (d) Image composite.

## 2.2 Projection du bloc

Dans cette partie, on explique la méthode de projection d’un bloc 2D vers des segments 1D afin d’étudier les variations le long de différentes directions. Dans le domaine discret, une direction est représentée par des droites discrètes dont l’épaisseur (entre autres) est variable. Revelles propose une description théorique de ces objets dans [3]. Notre algorithme consiste à projeter tous les pixels du bloc le long de droites discrètes 8-connexes (ou naïves), comme illustré dans la figure 2. Le choix de cette méthode de projection a été évalué dans [4] : notre méthode d’estimation de direction de contours est comparée à deux méthodes existantes, la transformée de Radon et la projection utilisée pour la construction de bases de bandelettes. Elle s’avère plus précise lors de l’estimation de direction de contours dans de petits blocs (à partir de blocs  $(8 \times 8)$  pixels et en-dessous), sur des contours bruités (bruit blanc et effet de blocs), et sur des images naturelles en général. Notons que le nombre de directions le long desquelles le bloc est projeté augmente avec la taille du bloc. En effet, plus le bloc est grand plus il est possible de définir des droites discrètes naïves distinctes, et plus la résolution angulaire est bonne. La relation exacte entre la taille du bloc et le nombre de directions que l’on peut créer dans ce bloc est définie mathématiquement par les suites de Farey.

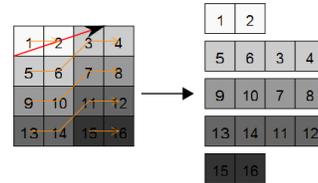


FIGURE 2 – Création de cinq segments 1D à partir d’un bloc  $(4 \times 4)$  pixels, le long de la direction de paramètres  $(1,3)$ .

## 2.3 Calcul des variations

Dans le but de trouver la direction prédominante de chaque bloc, les variations des segments projetés sont étudiées. La valeur de variation  $V_i$  du  $i$ -ème segment projeté  $s_i$  est définie par :

$$V_i = \sup(s_i) - \inf(s_i)$$

La variation globale  $V_{tot}$  de la direction  $\theta$  est calculée en moyennant les variations de chaque segment, avec  $I$  le nombre total de segments projetés :

$$V_{tot} = \frac{\sum_{i=1}^I V_i}{I}$$

L'angle associé  $\theta_{bloc}$  est celui qui correspond à la valeur de variation minimale, et que l'on estime parallèle au contour du bloc. La figure 3 illustre un cas où le bloc contient une direction de contour.

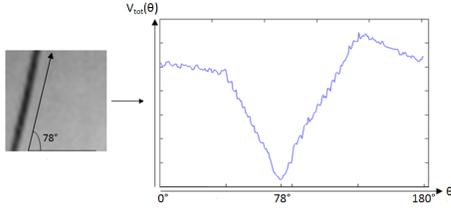


FIGURE 3 – Un bloc et sa courbe de variations. La valeur minimale correspond à un angle de  $78^\circ$ .

Dans le cas où plusieurs directions de contours sont présentes à l'intérieur du même bloc, ce dernier est divisé grâce à l'algorithme de *quad-tree*. Le critère qui détermine la présence d'une ou plusieurs directions dans le bloc est basé sur la comparaison entre les variations des coefficients le long de la direction  $\theta_{bloc}$ , et la variation totale des coefficients du bloc. Lorsqu'une division du bloc est nécessaire, quatre sous-blocs de tailles égales sont créés. Le calcul de direction prédominante est ensuite effectué à l'intérieur de chaque sous-bloc. Cet algorithme est itéré jusqu'à ce que tous les blocs contiennent au plus une direction de contours ou lorsqu'une taille minimale de bloc est atteinte ( $4 \times 4$  pixels). Un exemple est illustré en figure 4.

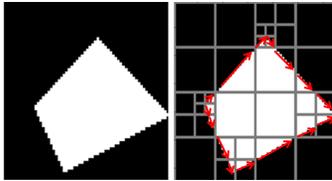


FIGURE 4 – Exemple de division en *quad-tree*.

### 3 Interpolation

Un certain nombre d'interpolations directionnelles [5][6][7] présentent des artefacts rédhibitoires car elles interpolent l'image entière en se basant uniquement sur la direction locale qu'elles ont préalablement détectée. Pour éviter de telles dégradations, notre méthode combine une interpolation isotrope (spline cubique) et une interpolation directionnelle. Cette dernière n'est utilisée que sur les contours dont l'orientation a été détectée lors de la phase précédente. Notons que des approches hybrides ont également été récemment proposées par Wang [8], Mallat [9] et Mueller [10]. Dans notre approche, la sélection des pixels

devant être interpolés directionnellement est obtenue avec des filtres de Gabor. La sortie de ce filtrage est utilisée comme un masque et pondère l'interpolation isotrope et l'interpolation directionnelle comme expliqué ci-dessous.

#### 3.1 Filtrage de Gabor

Les filtres de Gabor sont des filtres orientés passe-bande. Les paramètres d'un filtre sont l'échelle de l'enveloppe Gaussienne ( $\sigma_a, \sigma_b$ ), l'angle du filtre, la fréquence et la phase de la sinusoïde ( $\omega_0, P_0$ ). Afin de localiser les pixels qui doivent être interpolés directionnellement, les filtres de Gabor sont appliqués sur l'image d'ondelettes de détails à la résolution la plus fine, pour chaque bloc résultant du *quad-tree*. Les filtres sont orientés dans la direction attribuée au bloc, et les paramètres de l'enveloppe ainsi que la fréquence de la sinusoïde sont fixés empiriquement de sorte que la bande passante du filtre soit adaptée à la fréquence des coefficients d'ondelettes. Le fait d'appliquer ces filtres sur les coefficients d'ondelettes permet de filtrer des images dont le contenu fréquentiel varie peu, et donc de fixer de manière optimale les coefficients des filtres de Gabor sans que ceux-ci ne doivent être modifiés en fonction de l'image. La sortie de ce filtrage est un masque appelé  $M$  qui a une valeur élevée pour les pixels ayant la même orientation que le filtre, et faible pour les autres.

#### 3.2 Lissage Gaussien directionnel

L'interpolation directionnelle consiste à corriger l'interpolation spline du bloc en la filtrant avec un filtre Gaussien 2D orienté dans la direction  $\theta_{bloc}$  du contour du bloc, pour donner un bloc interpolé directionnellement  $B_{dir}$ . Les paramètres de la Gaussienne ( $\sigma_\theta$  et  $\sigma_{\theta^\perp}$ ) sont choisis tels que la Gaussienne soit allongée selon  $\theta$  et très fine dans la direction perpendiculaire  $\theta^\perp$  :  $\sigma_\theta \gg \sigma_{\theta^\perp}$ . Des exemples de filtres sont donnés en figure 5 pour plusieurs valeurs de  $\sigma_{\theta^\perp}$ . Ces paramètres permettent de créer des filtres dits *fins*, dont le nombre de coefficients non nuls est faible. Seuls les coefficients alignés dans la direction choisie sont non-nuls et permettent la reconstruction des contours orientés dans la même direction que le filtre, et l'élimination des artefacts. De plus, la finesse de ces filtres permet de ne pas introduire de flou lorsqu'ils sont appliqués.

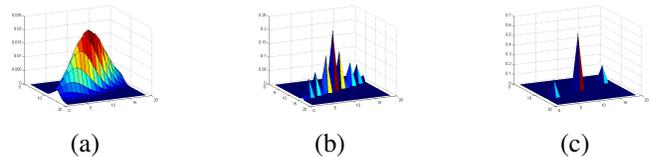


FIGURE 5 – Des filtres Gaussiens pour différentes valeurs du paramètre  $\sigma_{\theta^\perp}$ . (a)  $\sigma_{\theta^\perp} = 1.5$ . (b)  $\sigma_{\theta^\perp} = 0.15$ . (c)  $\sigma_{\theta^\perp} = 0.05$ .

### 3.3 Combinaison des interpolations isotrope et directionnelle

Afin de ne pas dégrader les pixels qui ne seraient pas orientés dans la direction  $\theta_{bloc}$ , seuls les pixels pour lesquels la sortie du filtre de Gabor est élevée sont lissés. Le schéma d'interpolation global est alors une combinaison linéaire du bloc interpolé par un noyau spline  $B_{spline}$  et du bloc filtré  $B_{dir}$ , pondérée par le masque  $M$  interpolé (par un noyau spline) et normalisé entre 0 et 1. Ainsi pour chaque pixel de coordonnées  $(x, y)$  :

$$B(x, y) = B_{dir}(x, y) \times M(x, y) + B_{spline}(x, y) \times (1 - M(x, y))$$

La figure 6 illustre le fonctionnement global de l'interpolation. Notons qu'afin d'éviter l'apparition d'un effet de bloc, un recouvrement est introduit lors des différents filtrages.

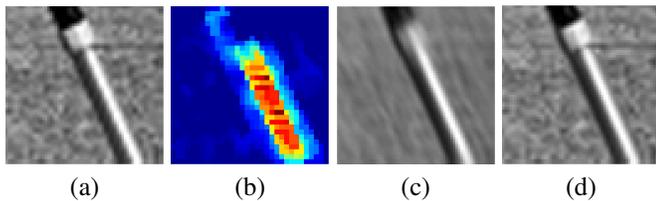


FIGURE 6 – (a) Bloc interpolé par un noyau spline. (b) Masque créé par le filtrage de Gabor. (c) Bloc (a) filtré par le filtre Gaussien. (d) Résultat final de la pondération.

## 4 Résultats

La figure 7 montre une série d'agrandissements d'un facteur  $2 \times 2$  avec quatre méthodes d'interpolation directionnelle récentes : la méthode SME de Mallat et Yu [9], celle de Wang et al (NOAI) [8], l'interpolation de NEDI de Li [5], et la nôtre (GCI, pour *Gaussian Corrected Interpolation*).

Notre méthode parvient à éliminer les artefacts produits par l'interpolation spline (*jaggy* et *aliasing*) sans en introduire de nouveaux (faux pixels, faux contours). Le comportement est également correct quand de nombreuses directions de contours sont présentes et lors de contours courbes. Des résultats quantitatifs (PSNR) ont été calculés [4] et montrent un léger gain de  $0.3dB$  en moyenne par rapport à l'interpolation NEDI, sur neuf images traitées. Une évaluation subjective des résultats permettrait de mieux conclure.

Au niveau des temps de calcul, notre méthode est moins rapide que l'interpolation NEDI mais environ dix fois plus rapide que la méthode SME pour une image  $256 \times 256$ , et vingt fois pour une image  $512 \times 512$  (interpolation d'un facteur  $2 \times 2$ ). De plus, plusieurs compromis peuvent être réalisés pour améliorer la vitesse de notre algorithme. Le nombre d'angles détectables peut être réduit, et la définition en pré-traitement d'un dictionnaire de filtres de Gabor et Gaussien améliorerait considérablement les temps de calcul sans que la qualité de l'interpolation ne soit altérée de manière significative.

Enfin, notons que les paramètres de nos filtres sont fixés automatiquement pour permettre un comportement optimal de l'al-

gorithme, sans que l'utilisateur n'intervienne. Cela permet une grande souplesse d'utilisation pour tous types d'images.

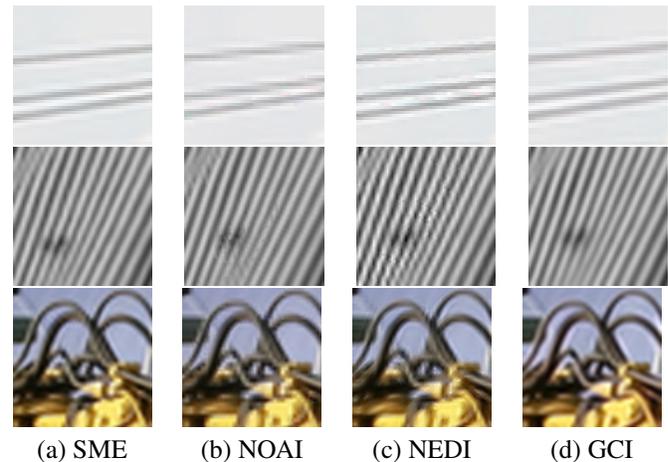


FIGURE 7 – Comparaison d'interpolations  $2 \times 2$  (voir la version électronique pour une meilleure visualisation)

## Références

- [1] G. Peyre and S. Mallat, "Discrete bandelets with geometric orthogonal filters," *ICIP*, vol. 1, pp. I – 65–8, sep. 2005.
- [2] J.L. Starck, J. Fadili, and F. Murtagh, "The undecimated wavelet decomposition and its reconstruction," *ICIP*, vol. 16, no. 2, pp. 297–309, Feb. 2007.
- [3] Jean-Pierre Reveilles, *Geometrie discrete, Calcul en nombres entiers et algorithmique*, Ph.D. thesis, Université Louis Pasteur, Strasbourg, France, 1991.
- [4] E. Van Reeth, *Système avancé d'interpolation spatiale de signaux de télévision pour affichage sur écrans haute-définition*, Ph.D. thesis, Université de Grenoble, 2011.
- [5] Xin Li and M.T. Orchard, "New edge-directed interpolation," *ICIP*, vol. 10, no. 10, pp. 1521–1527, 2001.
- [6] Hao Jiang and C. Moloney, "A new direction adaptive scheme for image interpolation," *ICIP*, vol. 3, pp. III–369 – III–372 vol.3, 2002.
- [7] D. Muresan, "Fast edge directed polynomial interpolation," *ICIP*, vol. 2, pp. II– 990–993, 2005.
- [8] Qing Wang and R.K. Ward, "A new orientation-adaptive interpolation method," *ICIP*, vol. 16, no. 4, pp. 889–900, april 2007.
- [9] S. Mallat and G. Yu, "Super-resolution with sparse mixing estimators," *IEEE Transactions on Image Processing*, vol. 19, pp. 2889–2900, 2009.
- [10] N. Mueller, Yue Lu, and Minh N. Do, "Image interpolation using multiscale geometric representations," *Computational Imaging*, vol. 6498, pp. 64980A, 2007.