

Utilisation d'un algorithme d'interpolation de données irrégulièrement réparties pour l'estimation et la compensation de mouvement dans une séquence d'images

Hervé Le Floch, Claude LABIT

IRISA/INRIA-Rennes,
Campus universitaire de Beaulieu
35042 Rennes Cedex, France

RÉSUMÉ

Dans ce papier, nous présentons une nouvelle modélisation du mouvement. Cette modélisation est issue d'une technique d'interpolation à partir d'échantillons irréguliers. Elle permet à partir d'un nombre limité de vecteur-échantillons de construire un champ dense de vecteurs. L'optimisation des paramètres liés aux vecteur-échantillons conduit à un champ dense de vecteurs minimisant l'erreur de compensation. Des résultats en terme de qualité de compensation en mouvement et en terme de codage sont donnés.

ABSTRACT

In this paper, a new motion model is presented. It results of a scattered data interpolation method. This model allows, from a fixed number of vectors, to build a dense vector motion field. Vector parameters optimization leads to a dense vector motion field minimizing the motion compensation error. Some results showing the motion compensation capacities and showing the compression capacities of this new modelisation are given.

1 Introduction

En raison des artéfacts connus des schémas classiques de compensation en mouvement (découpage en blocs et modèle de mouvement translationnel), de nouveaux modèles de mouvement et de partitionnements irréguliers sont apparus[1,5,7]. Ce papier propose l'utilisation d'un autre modèle de mouvement. Ce modèle de mouvement correspond à une extension 2D d'un algorithme d'interpolation de surface à partir d'échantillons irrégulièrement répartis[3,4]. Dans notre cas, chaque composante du champ de vecteur est traitée séparément et à partir d'un ensemble de vecteur-échantillons irrégulièrement répartis, un champ dense de vecteurs peut être obtenu et utilisé lors d'une phase de compensation en mouvement. L'objectif est donc d'optimiser les paramètres liés à chaque vecteur-échantillon (Dans notre cas, ces paramètres sont au nombre de trois) de façon à minimiser l'erreur de compensation. Nous présentons tout d'abord l'algorithme d'interpolation utilisé (interpolation de Shepard [2,6]). Nous présentons ensuite l'algorithme d'estimation des paramètres des vecteur-échantillons. Nous présentons enfin plusieurs schémas d'utilisation de cet algorithme d'estimation ainsi qu'une comparaison avec un autre algorithme d'estimation/compensation de mouvement (modélisation affine sur un treillis triangulaire).

2 L'algorithme d'interpolation

Nous utilisons une version modifiée de l'algorithme de Shepard [6]. Cet algorithme permet l'interpolation d'une surface à partir d'une grille d'échantillons irrégulièrement

répartis dans le domaine image. La reconstruction d'un pixel manquant s'obtient en deux étapes :

- sélection des N échantillons les plus proches du pixel.
- reconstruction du pixel par pondération des N échantillons sélectionnés par une fonction dépendant de la distance entre pixel et échantillons.

L'interpolation d'un pixel localisé en (x,y) à partir des N échantillons les plus proches est donné par :

$$\hat{F}(x, y) = \sum_{i=1}^M W_i(x, y) \cdot f_i \quad (1)$$

avec :

$$W_i(x, y) = \frac{\left(\frac{(R(x,y,N) - d_i(x,y))_+}{R(x,y,N) \cdot d_i(x,y)} \right)^{\psi_i(x_i, y_i)}}{\sum_{k=1}^N \left(\frac{(R(x,y,N) - d_k(x,y))_+}{R(x,y,N) \cdot d_k(x,y)} \right)^{\psi_k(x_k, y_k)}} \quad (2)$$

avec :

$$(R - d)_+ = \begin{cases} R - d & \text{if } R - d \geq 0 \\ 0 & \text{else} \end{cases} \quad (3)$$

$R(x,y,N)$ représente la distance entre le dernier échantillon sélectionné et le pixel interpolé. $d_i(x, y)$ est la distance entre le pixel interpolé et l'échantillon numéro i . A chaque échantillon sont associés 3 paramètres : sa position (x_i, y_i) , sa valeur (f_i) et un terme d'influence de l'échantillon ψ_i (un ψ_i élevé conduit à une large zone plate autour de l'échantillon). Dans la méthode de Shepard initiale, les valeurs des ψ_i sont fixées à 2. Cet algorithme présente plusieurs propriétés intéressantes :

- l'algorithme d'interpolation est local.

- Les échantillons peuvent être localisés en tout point de l'image. La localisation d'échantillons sur les coins ou les bords de l'image n'est pas nécessaire (propriété de "non convexité" de l'algorithme de Shepard).

Dans [3,4], nous appliquons l'interpolation de Shepard au sous-échantillonnage adaptatif d'images fixes. Un processus d'insertion d'échantillons est suivi d'une optimisation des paramètres des échantillons par un algorithme de relaxation. Dans [4], nous avons ajouté un paramètre permettant d'introduire un terme d'anisotropie dans le schéma d'interpolation. Nous approximons efficacement la surface photométrique d'images. Un codage de la carte d'échantillonnage par un codage arithmétique et une quantification des valeurs des échantillons conduit à des taux de compression intéressants. Cet algorithme amène un positionnement des échantillons de part et d'autre des contours de l'image. Aucune structure de graphe n'est utilisée et n'est donc à transmettre. La simple transmission des paramètres des échantillons suffit à la reconstruction. Les positions des échantillons sont des paramètres directs de l'algorithme de minimisation et peuvent être déplacés très précisément lors du processus de relaxation.

3 Extensions au cas 2D

Nous présentons dans ce paragraphe l'extension de l'algorithme d'approximation d'images fixes à l'estimation et la compensation de mouvement dans une séquence d'images.

3.1 Construction d'un champ de vecteurs dense

Le passage de la surface photométrique à un champ de vecteurs est immédiat. Chaque composante d'un vecteur est considérée indépendamment et est interpolée selon la loi donnée par l'algorithme d'interpolation modifiée de Shepard. Si $\vec{d}_i(x_i, y_i) = (d_{xi}, d_{yi})^T$, avec $i=1\dots M$, est un ensemble de M vecteurs localisés en (x_i, y_i) , le vecteur reconstruit en un pixel quelconque de l'image est donné par :

$$\vec{d}(x, y) = \begin{pmatrix} d_x(x, y) \\ d_y(x, y) \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^M W_{xi}(x, y) d_{xi}(x_i, y_i) \\ \sum_{i=1}^M W_{yi}(x, y) d_{yi}(x_i, y_i) \end{pmatrix}$$

W_{xi} est le poids accordé à la composante horizontale du vecteur d_i , W_{yi} est le poids accordé à la composante verticale du vecteur d_i . Leurs valeurs sont données par la formule (2). Leurs valeurs peuvent être différentes du fait de la présence du paramètre $\psi = (\psi_x, \psi_y)$. La compensation d'une image I_k s'exprime donc en tout pixel de l'image par :

$$\hat{I}_k(x, y) = \hat{I}_{k-1}(x + d_x(x, y), y + d_y(x, y)) = \hat{I}_{k-1}(x', y')$$

Lorsque (x', y') tombe sur une position non entière, la valeur de l'intensité est calculée par interpolation bilinéaire à partir des 4 plus proches voisins.

3.2 Application à l'estimation et à la compensation de mouvement

L'objectif est de trouver, pour un budget donné de vecteurs-échantillons, les paramètres minimisant l'erreur de compen-

sation. Les valeurs des ψ_i sont discrétisées et se limitent à 1,2 ou 3. L'algorithme d'optimisation se décompose en deux étapes :

- **Insertion** d'un nombre fixe de vecteurs-échantillons et initialisation des valeurs de ces vecteurs par block-matching. Deux méthodes d'insertion peuvent être envisagées : soit une insertion directe sur les structures principales de l'image, soit une insertion progressive, avec, après chaque insertion, une interpolation locale du champ de vecteurs et une compensation locale. Dans ce dernier cas, les vecteur-échantillons sont insérés sur les zones de plus fortes erreurs de compensation. Lors de cette phase d'insertion, la valeur initiale des ψ est fixé à 2 (composantes horizontales et verticales). Lors de l'insertion, la densité des vecteur-échantillons est limitée de façon à éviter les trop fortes concentrations de vecteurs.
- **Relaxation** sur les paramètres des vecteur-échantillons. Chaque vecteur-échantillon est successivement sélectionné, il se déplace vers la position adjacente minimisant l'erreur de compensation puis adopte la valeur voisine minimisant l'erreur de compensation. Après l'obtention de la position et valeur du vecteur-échantillon, l'ensemble des valeurs des ψ sont testées. Celle minimisant l'erreur de compensation est retenue. Cette phase s'avère possible par utilisation de la propriété de localité de l'algorithme de Shepard. Le processus de relaxation est répétée jusqu'à convergence totale.

Remarque : Dans ce schéma d'optimisation, les composantes horizontales et verticales des vecteur-échantillons pourraient être traitées de façon totalement indépendantes. Dans ce cas, les positions des vecteur-échantillons seraient différentes selon qu'elles représentent la composante horizontale ou verticale du champ de vecteurs.

3.3 Utilisation dans un schéma d'estimation/compensation de mouvement

L'utilisation d'un tel modèle de mouvement dans un schéma de codage peut se faire selon une hiérarchie d'optimisation. Plusieurs cas sont envisageables (selon le niveau d'information que l'on a à transmettre).

1. Optimisation totale de l'ensemble des paramètres (Position/valeur/influence).
2. Optimisation valeur et influence.
3. Optimisation valeur.

Le cas 1 nécessite la transmission des positions des vecteur-échantillons alors que les cas 2 et 3 ne le nécessite pas. Dans les cas 2 et 3, 4 possibilités peuvent être envisagées :

- Les vecteur-échantillons sont positionnés sur une grille régulière.
- Les vecteur-échantillons pointent sur les structures importantes (par exemple, gradients maximaux) de l'image précédente (image présente au décodeur)[1]. Le processus d'ajustement simultané des positions et valeurs sera contraint par cette condition.

- Les vecteur-échantillons sont insérés sur les forts gradients de la première image. Leurs positions sont ensuite suivies au cours du temps le long de la séquence à partir des valeurs des vecteur-échantillons (avec une réinsertion automatique des vecteur-échantillons pour éviter une trop forte concentration de vecteur-échantillons).
- L'ensemble des paramètres des échantillons sont optimisés sur le couple d'images précédentes présentes au décodeur (estimation des meilleures positions et valeurs entre I_{k-1} et I_{k-2}). Ces positions sont ensuite projetées sur l'image I_k . Les paramètres valeur et influence sont ensuite optimisés sur I_k .

4 Résultats et comparaisons

Dans ce paragraphe, nous donnons les résultats obtenus avec les schémas d'estimation 1,2 et 3 du paragraphe précédent. Dans un premier temps, nous donnons les résultats uniquement en terme de capacité de compensation de mouvement (Mesure de l'erreur quadratique de compensation sans prise en compte du cout de codage de la modélisation utilisée). Puis nous présentons des résultats obtenus dans un schéma de codage de séquences d'images. Dans les 2 cas, nous comparons nos résultats à ceux obtenus avec un algorithme du type triangulation (de Delaunay) et modélisation par mouvement affine à 6 paramètres de la déformation de chaque triangle. La comparaison avec un algorithme de Block-matching (full-search) n'est pas montrée car les résultats sont inférieures à ceux obtenus par les algorithmes testés (triangulation ou méthode de Shepard)

Le schéma basé sur la triangulation est une version simplifiée de celui présenté dans [1]. Dans [1], afin d'éviter la transmission des positions des vecteurs, ceux-ci devaient pointer sur certains pixels (zone de forts gradients ou d'erreurs de compensation importantes) de l'image précédente (image présente au décodeur). Dans notre cas, ces points seront choisis sur l'image courante (Nous supposons cependant que leurs positions ne sont pas à transmettre). Ainsi, seules les valeurs des vecteurs seront optimisées. L'algorithme d'optimisation utilisé est le même que celui utilisé dans notre algorithme : Choix de points de fort gradient avec limitation de la densité de vecteurs, insertion de vecteurs sur ces positions et initialisation de la valeur du vecteur par block-matching, processus de relaxation sur les valeurs des vecteurs par test et choix de valeurs de vecteurs voisines. Du fait de la propriété de convexité de la triangulation, 1 vecteur est inséré à chaque coin de l'image. Notre algorithme de triangulation ne réalise pas un contrôle du respect de la topologie. Cette comparaison n'est donc pas une comparaison avec la méthode présentée dans [1], mais nous pensons qu'elle peut amener un premier élément de comparaison intéressant. Les expérimentations présentées dans ce papier ont été menées sur la séquence QCIF "Suzie". Les expériences ont été menées sur les images 25 à 55. Cette séquence présente un mouvement de faible amplitude sur les premières images et un mouvement complexe de forte amplitude par la suite.

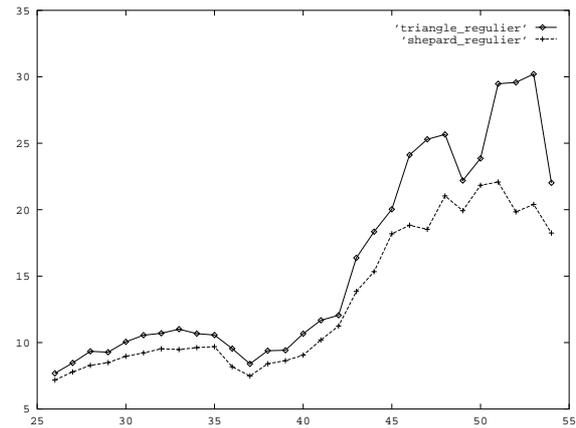


FIG. 1 — EQM obtenue sur les images 25 à 55 de la séquence Suzie. Méthode de type Triangulation et de type Shepard. 120 vecteurs régulièrement répartis

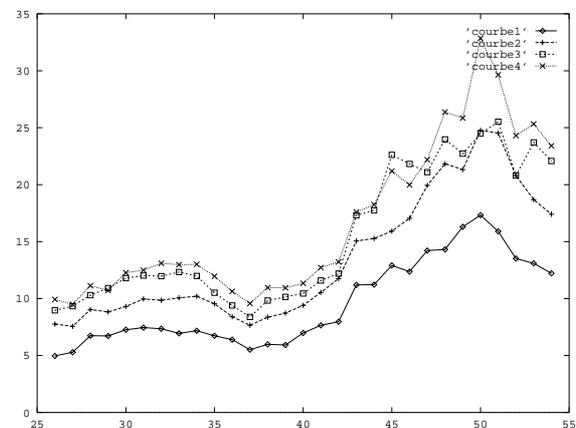


FIG. 2 — EQM obtenue sur les images 25 à 55 de la séquence Suzie. Méthode de type Triangulation (courbe 3) et de type Shepard (ajustement valeur/position/influence : courbe 1, ajustement valeur/influence : courbe 2, ajustement valeur : courbe 4). 99 vecteurs régulièrement répartis

4.1 Estimation/compensation du mouvement

La figure 1 montre les résultats obtenus (en terme d'Erreur Quadratique Moyenne) lorsque 120 vecteur-échantillons sont positionnés sur une grille régulière. Dans le cas de la triangulation, les valeurs de vecteurs sont optimisées, dans le cas de l'algorithme de Shepard, les valeurs des vecteurs et les paramètres d'influence sont optimisés. Dans les 2 cas, les vecteurs sont identiquement positionnés et identiquement initialisés. Ces résultats montrent que, indépendamment du cout de codage, l'algorithme basé sur l'interpolation de Shepard permet d'obtenir une meilleure qualité de compensation que celle obtenue par l'algorithme de type triangulation. La figure 2 compare les méthodes suivantes (99 vecteurs) :

1. Méthode de Shepard avec ajustement des paramètres de position, de valeur et d'influence (courbe 1).
2. Méthode de Shepard avec ajustement des paramètres de valeur et d'influence. Les vecteur-échantillons pointent

sur des pixels de fort gradient déterminés sur l'image précédente (courbe2).

3. Méthode basée sur la triangulation et la modélisation du mouvement de chaque triangle selon un modèle affine. Les vecteurs sont positionnés sur les positions de fort gradient de l'image courante (courbe3).
4. Méthode de Shepard avec ajustement des paramètres de valeur. Les vecteur-échantillons pointent sur des pixels de fort gradient déterminés sur l'image précédente (courbe4).

Les résultats de la figure 2 montrent que l'ajustement des paramètres de position apportent une réelle amélioration quant à la qualité de compensation en mouvement. L'optimisation des paramètres d'influence permet d'obtenir de meilleurs résultats que ceux obtenus par triangulation.

4.2 Application au codage d'images

Le surcout de la méthode basée sur l'algorithme de Shepard (par rapport à une méthode de triangulation) est dû à la transmission des positions des échantillons et à la transmission des paramètres d'influence. Nous avons construit un code de Huffman basé sur les probabilités d'apparition des couples (ψ_x, ψ_y) . Le cout des paramètres ψ , avec le code élaboré, est d'environ 3 bits par vecteur-échantillon sur la séquence testée. Le cout de la carte binaire des positions des vecteur-échantillons par un codage arithmétique avec modèle fixe [8] est de 960 bits par image. Nous donnons dans ce papier les résultats obtenus avec un schéma simple de codage de séquence d'images : prédiction de l'image courante et transmission de l'image d'erreur par JPEG. Parmi les prédicteurs en mouvement testés, nous avons utilisé l'algorithme de compensation par triangulation (en supposant que les positions des vecteurs ne sont pas à transmettre) (méthode 1), l'algorithme par méthode de Shepard dans lequel les positions, les valeurs et les paramètres d'influence sont optimisés (méthode 2), l'algorithme de Shepard dans lequel les valeurs et les paramètres d'influence (méthode 3) sont optimisés (les vecteur-échantillons pointent vers des pixels de fort gradient calculés sur l'image précédente décodée, les positions des vecteur-échantillons ne sont donc pas à transmettre). Le tableau ci-dessous donne le PSNR (dB) moyen obtenu sur les 30 images ainsi que le débit (bits par pixel). Dans ces 3 méthodes, aucun codage n'a été effectué sur les valeurs des vecteurs, le cout de codage des valeurs des vecteurs est supposé être le même pour les 3 méthodes testées.

No	PSNR	Débit	PSNR	Débit	PSNR	Débit
1	36.82	0.287	34.74	0.203	38.06	0.394
2	37.26	0.296	35.35	0.252	38.49	0.392
3	36.79	0.286	34.81	0.207	37.99	0.393

Ce tableau montre qu'en terme de débit/distorsion, l'optimisation des paramètres de position, de valeur et d'influence présente tout son intérêt pour des débits élevés. A "bas" débit, le surcout d'information du aux transmissions des paramètres de position et d'influence devient pénalisant.

5 Conclusion et perspectives

Nous avons proposé une nouvelle modélisation du mouvement dans une séquence d'images issue d'un algorithme d'interpolation à partir de données irrégulièrement réparties. Cette modélisation permet l'optimisation de 3 niveaux de paramètres et ne nécessite aucune contrainte de respect de topologie ou de respect de convexité. Nous travaillons actuellement à l'amélioration de la modélisation par introduction d'un paramètre d'anisotropie et à l'utilisation de cette algorithme d'estimation du mouvement dans le cadre de l'interpolation temporelle. Nous espérons ainsi utiliser au maximum les qualités d'estimation en mouvement du modèle présenté.

Références

- [1] M. Dudon, *Modélisation du Mouvement par Treillis Actifs et Méthodes d'estimation associées. Application au codage de séquence d'images*, Thèse de l'université de Rennes1.
- [2] W. J. Gordon and J. . Wixom *Shepard's Method of Metric Interpolation to Bivariate and Multivariate Interpolation*, in *Mathematics of Computation*, Vol.32, no.4, pp. 253–264, January 1978.
- [3] H. Le Floch, C. Labit *A Scattered Data Interpolation Algorithm for Still-Image Sub-sampling and for Motion Field Representations used to Video Coding*, SPIE Visual Communications and Image Processing 1997, Vol. 3024, pp. 635–646.
- [4] H. Le Floch, C. Labit, *Irregular Image Sub-sampling and Reconstruction by interpolation*, Proceedings de SAMPTA-97 (Sampling Theory and Application), June 1997.
- [5] Y. Nakaya, H. Harashima, *Motion Compensation Based on Spatial Transformations*, IEEE Transactions on Circuits and Systems for Video. June 1994, vol. 4, number 3, pp 339-356.
- [6] D. Shepard, *A two Dimensional Interpolation Function for Irregularly Spaced Data*, Proc. 23rd Nat. Conf. ACM, 1968, pp 517-523.
- [7] Y. Wang, O. Lee, *Active Mesh- A Feature Seeking and Tracking Image Sequence Representation Scheme*, IEEE Transactions on Image Processing. Septembre 1994, vol. 3, number 5, pp 610-624.
- [8] I.A. Witten, R.M. Neal, J. G. Cleary, *Arithmetic Coding For Data Compression*, in *Communication on ACM*, June 1987, vol. 30, number 6, pp. 520–540.