Utilisation des distances tangentes pour la compensation de mouvement

J. Fabrizio[1] S. Dubuisson[1]

[1]Laboratoire d'Informatique de Paris 6 (UMR CNRS 7606)
Université Pierre et Marie Curie-Paris6

104 avenue du Président Kennedy, 75015 Paris

{Jonathan.Fabrizio,Severine.Dubuisson}@lip6.fr

Résumé

Dans cet article, nous présentons un algorithme de compensation de mouvement qui repose sur l'utilisation des distances tangentes. Contrairement aux algorithmes de Block-Matching classiques, qui prédisent uniquement l'évolution de la position spatiale de blocs image, notre algorithme prédit aussi l'évolution des pixels au cours du temps à l'intérieur même de ces blocs. De ce fait, l'erreur de prédiction diminue fortement. Intégré dans une chaîne de compression complète, notre algorithme pourrait en améliorer le taux de compression et la qualité de reconstruction. Le nombre d'images de référence nécessaires à l'encodage d'une séquence peut aussi être diminué par l'utilisation de notre algorithme.

Mots clefs

Distance tangente, Compensation de mouvement, Codage, Compression.

1 Introduction

La nécessité d'accroître la résolution ou la qualité des séquences vidéo nous oblige à améliorer les méthodes de compression. Dans une séquence vidéo, deux images successives sont supposées être relativement similaires. Les seuls changements observés sont dus aux mouvements des objets et/ou de la caméra. Le principe de base de la compression vidéo est de prédire une nouvelle image à partir d'une autre, et de ne coder que les erreurs de prédictions. En effet, ces erreurs ont des énergies plus petites que les valeurs originales des pixels : leur codage nécessite donc moins de bits puisque l'entropie est plus faible. Il existe deux types de prédictions : la prédiction temporelle et la prédiction spatiale.

Pour la prédiction temporelle, ou inter-images, considérant un pixel courant, on fait l'hypothèse que ses voisins dans les images antérieures et postérieures lui sont similaires. L'estimation de mouvement est largement utilisée pour l'élimination de la redondance temporelle entre les images d'une séquence vidéo [1]. On peut ainsi faire de la prédiction par compensation de mouvement : (i) de manière unidirectionnelle (prédiction avant ou arrière), sans prise en compte des problèmes d'occultation, ou (ii) de manière bidirectionnelle, en utilisant à la fois les informations antérieures et postérieures dans la séquence. La plupart des standards de compression vidéo, tels que MPEG [2], utilisent des méthodes dites de *Block-Matching*. Le principe est de réaliser une partition de l'image en blocs et de rechercher la position optimale de ces blocs dans une autre image, selon un critère de maximisation de corrélation. De nombreux algorithmes de *Block-Matching* existent. On peut trouver de bonnes synthèses dans les articles [3, 4, 5] ainsi qu'une étude empirique comparative dans [6].

La prédiction spatiale vise à considérer les pixels adjacents à l'intérieur des images (ou intra-images) : pris dans l'ordre lexicographique du tampon image, ces pixels sont supposés similaires. En considérant les pixels précédemment parcourus (donc codés), dans un certain voisinage, l'encodeur peut prédire la valeur du pixel courant. Les principales problématiques posées sont le choix du voisinage et des coefficients pondérateurs à affecter aux valeurs des pixels de ce voisinage. En général, cette prédiction spatiale s'adapte au contenu (frontières, régions, *etc.*). Il existe de nombreux prédicteurs spatiaux, parmi lesquels on peut citer le prédicteur adaptatif médian (*MAP*) [7] sur lequel est basé par exemple l'algorithme LOCO-I [8].

Dans cet article nous proposons une approche originale de compensation de mouvement. Cette approche repose sur un nouvel algorithme d'estimation de mouvement, basé sur les distantes tangentes [9]. Dans la section 2, nous ferons un rappel sur les distances tangentes puis, dans la section 3 nous présenterons l'algorithme de compensation de mouvement. Nous donnerons ensuite des résultats qualitatifs et comparatifs en section 4, avant de conclure en section 5.

2 La distance tangente

La distance tangente est un outil mathématique qui permet de comparer deux motifs (ou blocs) en tenant compte de petites transformations (rotations, translations, *etc.*). Introduite au début des années 90 par Simard *et al.* [10], elle a été combinée avec différents classifieurs pour la reconnaissance de caractères [11, 12, 13], la détection et recon-

naissance de visages [14] ainsi que la reconnaissance de la parole [15]. Elle est encore peu utilisée.
La distance d'un motif P à un autre motif P' est calculée en mesurant la distance entre les espaces paramétriques passant par P et P' respectivement. Ces espaces modélisent localement l'ensemble des formes engendrées par les transformations possibles entre les deux motifs (figure 1.B). Ils sont obtenus en linéarisant chaque transformation c'est-à-dire en ajoutant au modèle global les vecteurs tangents à ces transformations, pondérés par un coefficient. Chaque vecteur tangent dérive, en un point, l'ensemble des formes engendrées par une transformation.
Soit I_m le bloc obtenu après avoir appliqué une composition de transformations au bloc I. Il est possible d'approcher linéairement I_m en utilisant une expansion de Taylor :

$$I_m = I + \lambda_1 \vec{v_1} + \lambda_2 \vec{v_2} + ... + \lambda_n \vec{v_n} \quad (1)$$

où $\vec{v_i}$ est le vecteur tangent à la $i^{\text{ème}}$ transformation et λ_i correspond à l'apport de la $i^{\text{ème}}$ transformation. Dans notre cas, nous cherchons la transformation entre les blocs I et I_m : chaque vecteur tangent $\vec{v_i}$ dérive les variabilités locales engendrées par une transformation élémentaire (translation, rotation, changement d'échelle, *etc.*). L'espace formé par ces vecteurs, ou espace tangent, donne une représentation de tous les blocs, engendrés par la composition des transformations appliquées à I. La figure 1.A donne une représentation des vecteurs tangents correspondant à différentes transformations élémentaires.
Si l'on souhaite comparer deux blocs I_1 et I_2, on ne calculera donc pas simplement leur distance $d(I_1, I_2)$, mais on cherchera plutôt à minimiser celle entre les deux espaces tangents (figure 1.B) :

$$d(I_1+\lambda_{1,1}\vec{v_{1,1}}+...+\lambda_{1,n}\vec{v_{1,n}}, I_2+\lambda_{2,1}\vec{v_{2,1}}+...+\lambda_{2,n}\vec{v_{2,n}}) \quad (2)$$

où d est la distance euclidienne des niveaux de gris pixel à pixel. Dans cette expression, les blocs I_1 et I_2 sont connus. Les vecteurs tangents $\vec{v_{i,j}}$ correspondent à la dérivée en I_i de la $j^{\text{ème}}$ transformation et $\lambda_{i,j}$ est l'apport de la $j^{\text{ème}}$ transformation au $i^{\text{ème}}$ bloc. Les vecteurs tangents peuvent être calculés de la façons suivante :

$$I_t = I + \lambda_t \vec{v_t} \Rightarrow \vec{v_t} = \frac{I - I_t}{\lambda_t} \quad (3)$$

(le bloc I_t pouvant être généré de manière synthétique). Cette minimisation peut se faire numériquement, mais on trouve également une solution analytique dans [11]. Le résultat nous fournit donc, non seulement la distance minimale entre les deux blocs (en tenant compte des transformations à appliquer à I_1 pour aller vers I_2 et réciproquement), mais aussi les valeurs des $\lambda_{i,j}$ indiquant l'apport de chaque vecteur tangent, ce qui permet d'estimer la transformation réelle entre les deux blocs.

3 La compensation de mouvement

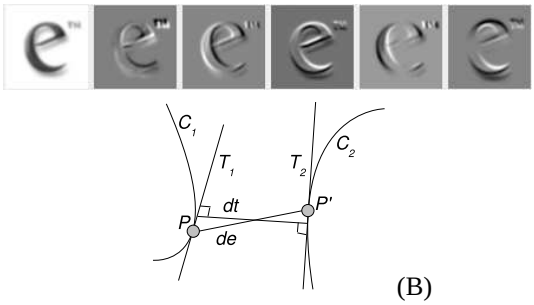
L'algorithme présenté ici repose sur l'utilisation de la distance tangente pour faire de la prédiction par compensation de mouvement. Notre but est, pour un bloc donné, de connaître son déplacement, c'est à dire, de trouver le bloc le plus ressemblant dans l'image d'arrivée, mais aussi de déterminer les transformations à l'intérieur de ce bloc. Chaque bloc I_2 de l'image d'arrivée est comparé avec des blocs I_1 spatialement proches dans l'image de départ. Pour cela, on minimise l'équation suivante :

$$d(I_1 + \lambda_{1,1}\vec{v_{1,1}} + ... + \lambda_{1,n}\vec{v_{1,n}}, I_2) \quad (4)$$

Au moment de l'encodage (algorithme 1), I_1 et I_2 sont connus, les vecteurs $\vec{v_{1,i}}$ sont calculés à l'aide de I_1 (équation 3). La minimisation de l'équation 4 nous permet de déterminer le bloc I_1 le plus similaire au bloc I_2 et les $\lambda_{1,i}$ qui donnent une estimation de la transformation à appliquer au bloc I_1 pour le mettre en correspondance avec le bloc I_2. Bien que I_2 soit connu au moment de l'encodage, ce n'est pas le cas au moment du décodage. Il faut donc utiliser l'équation 4 plutôt que l'équation 2 car il n'est pas possible de calculer les $\vec{v_{2,i}}$ à la décompression. Par ailleurs, l'encodage nécessite la minimisation de l'équation 4. Cette minimisation se résume à une inversion matricielle (pseudo-inverse), opération qui ne présente aucune difficulté et permet un encodage rapide.
Au moment du décodage (algorithme 2), l'image prédite est obtenue à partir de chaque bloc I_1 de l'image de départ, auxquels on a appliqué les vecteurs de déplacement et la transformation estimés lors de l'encodage. Pour cela, connaissant le bloc, on calcule les vecteurs tangents $\vec{v_{1,i}}$ puis, à l'aide des valeurs λ associées, on calcule un bloc I'_1 tel que :

$$I'_1 = I_1 + \lambda_{1,1}\vec{v_{1,1}} + ... + \lambda_{1,n}\vec{v_{1,n}} \quad (5)$$

L'algorithme est assez flexible car il est possible d'inclure dans le modèle différentes transformations (voire d'adapter les transformations modélisées à l'image). Ces transformations peuvent être géométriques (rotations, translations...) ou physiques (modifications d'illumination). Le



(A) (B)

Figure 1 – *(A). Vecteurs tangents à l'image d'origine (à gauche), avec, de gauche à droite : rotation, translation horizontale et verticale, facteur d'échelle horizontal et vertical. (B). Comparaison des motifs P et P'. La distance euclidienne est représentée par le segment $[de]$ et la distance tangente par le segment $[dt]$. Les transformations possibles de P et P' sont représentées par les courbes de l'espace C_1 et C_2 respectivement. Ces courbes sont respectivement linéarisées en P et P' pour donner T_1 et T_2 : leurs vecteurs directeurs sont les vecteurs tangents, et leur distance correspond à la distance tangente.*

Algorithme 1 Encodage

{calcul de la prédiction}
pour tout bloc b de l'image d'arrivée **faire**
  pour tout bloc b' proche dans l'image de départ **faire**
    Calcul des vecteurs tangents $\vec{v}$ du bloc b'
    Minimisation de $d(b' + \lambda v, b)$
    si le score de minimisation est plus faible que les précédents **alors**
      $b'_{\text{min}} \Leftarrow b'$
    finsi
  fin pour
  Calcul du déplacement entre le bloc b et le bloc b'_{min}
  Sauvegarde des vecteurs de déplacement et des λ qui minimisent $d(b'_{\text{min}} + \lambda v, b)$
fin pour
{calcul de l'erreur de prédiction}
Encodage de l'erreur de prédiction

Algorithme 2 Décodage

{génération de la prédiction}
pour tout bloc b de l'image d'arrivée **faire**
  Déterminer le bloc b'_{min} dans l'image de départ (selon les vecteurs de déplacement estimés et sauvegardés à l'encodage)
  Calculer les vecteurs tangents $\vec{v}$ au bloc b'_{min}
  Générer un bloc b''_{min} en utilisant les λ calculés et sauvegardés à l'encodage : $b''_{\text{min}} \Leftarrow b'_{\text{min}} + \lambda v$
  Le bloc b''_{min} est la prédiction du bloc b
fin pour
{erreur de prédiction}
Décodage de l'erreur de prédiction et correction de la prédiction

décodage se résume à l'évaluation de l'équation 5 : il est donc exécutable en temps réel.

Dans la section qui suit, nous donnons des résultats qualitatifs de prédiction par compensation de mouvement obtenus avec notre méthode. Nous les comparons avec ceux obtenus par simple *Block-Matching*.

4 Résultats

Puisque l'on estime une transformation à la fois spatiale et temporelle entre les blocs de deux images, nécessairement, l'erreur de prédiction, comparée à un celle d'un *Block-Matching* classique, diminue. Par conséquent, le taux de compression est amélioré. En revanche, la méthode impose, pour pouvoir décoder l'image prédite, de connaître les valeurs des λ. Cela nécessite donc leur enregistrement, au même titre que celui des vecteurs de déplacement des blocs. La question est de savoir si l'amélioration apportée par notre méthode, à savoir une diminution de l'erreur de prédiction, n'est pas pénalisée par l'ajout d'informations nécessaires que représentent les λ, et donc l'augmentation de la taille nécessaire au codage.

Pour répondre à cette question nous prédisons, à partir d'une image d'une séquence, l'image suivante et comparons les résultats obtenus avec ceux du *Block-Matching*. Le premier test porte sur les images (A) et (B) de la figure 2. Le mouvement réalisé entre ces images est un zoom, difficile à prédire du fait de la disparition d'objets.

Les images, en niveaux de gris et codées sur 1 octet, sont de taille 512×480 (soit 245760 octets). Pour la comparaison, nous considérons des blocs de taille 8×8 et une fenêtre de recherche de taille 23×23. Concernant notre algorithme, nous modélisons trois transformations : l'étirement horizontal, l'étirement vertical et un changement d'illumination. Davantage de transformations pourraient être envisageables, mais elles impliquent une augmentation de la quantité d'informations à sauvegarder. Les valeurs des différents λ sont arrondis à 10^{-1} pour faciliter leur encodage. Étudions la différence entre les images générées et l'image finale (erreur de prédiction). Les prédictions obtenues par les différentes méthodes sont visibles à la figure 4. Les différences entre l'image d'arrivée et l'image départ (sans) - (A), l'image prédite par *Block-Matching* (bm) - (B) et l'image prédite par les distances tangentes (dt) - (C), sont présentées par les histogrammes de la figure 3. Cette figure montre que notre méthode induit une faible amplitude des valeurs de l'erreur (valeurs proches de 0) ainsi qu'une forte redondance statistique. Ce double avantage réduit fortement la quantité d'informations à coder. Pour appuyer notre propos, quelques éléments statistiques concernant ces histogrammes sont également visibles dans le tableau 1.

	range	nb de valeurs	moyenne	variance	nb de zéro
sans	$[-254, 188]$	404	0.04	22.97	7866
bm	$[-199, 90]$	250	-0.64	$12,05$	21689
dt	$[-88, 70]$	125	0	$3,75$	67292

Tableau 1 – *Comparaison des histogrammes des erreurs de prédiction (figure 3) obtenues par les différentes techniques.*

	EQM	*PSNR*	entropie	taille théorique (bits)
sans	527.7	20.9	6.15	1511213
bm	145.5	26.5	5.21	1281287
dt	14.1	36.6	3.33	818557

(A)

	EQM	*PSNR*	entropie	taille théorique (bits)
sans	4919.5	11.2	8.0	1966139
bm	3094	13.2	7.5	1847711
dt	189.5	25.3	5.1	1243822

(B)

Tableau 2 – *Comparaison des prédictions obtenues sans estimation de mouvement (sans), avec estimation par Block-Matching (bm) et avec notre approche (dt), (A) si les deux images considérées sont consécutives dans la séquence, (B) si 12 images séparent les deux images.*

Le tableau 2.(A) montre clairement l'amélioration apportée par notre algorithme : diminution de l'erreur de prédiction et de son entropie. La taille théorique nécessaire pour coder l'erreur de prédiction est de 102319 octets auxquels il faut ajouter 3 valeurs par bloc de 64 pixels (pour les λ) soit environ 11520 octets, et ainsi un total de 113839 contre 160160 octets pour le *Block-Matching*. Ce calcul est théorique et basé sur l'entropie. Incluse dans une chaîne complète et utilisant des méthodes d'encodage plus poussées (*DCT*, transformée en ondelette,...) notre méthode donnerait des taux plus importants (erreur de prédiction plus faible, longues suites de zéros, *etc.*).

Pour bien prendre conscience de l'apport de notre méthode et surtout de sa robustesse, nous avons effectué

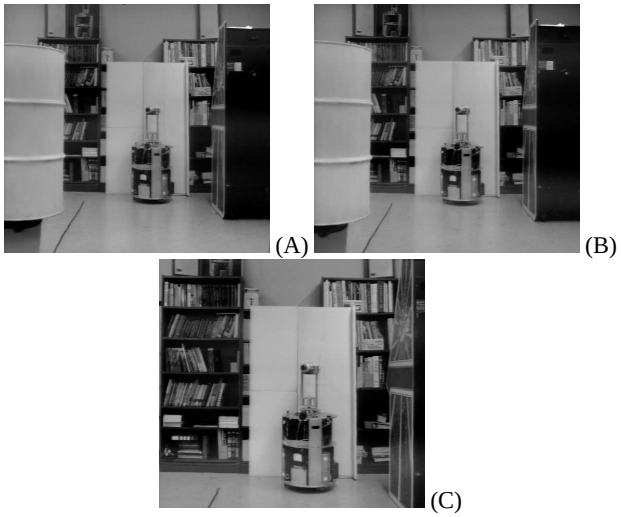


Figure 2 – *Séquence de travail (zoom)[16]. (A) : Image 0. (B) : Image 1. (C) : Image 12 (le zoom de la caméra a complètement fait disparaître une grande partie de l'image, dont le tonneau à gauche).*

cette comparaison dans une situation plus critique en passant 12 images entre l'image originale (figure 2.A) et l'image à prédire (figure 2.C - le tonneau de gauche disparaît). Les images prédites par les deux approches sont visibles sur la figure 5. Qualitativement, le *Block-Matching* est complètement mis en défaut (le tonneau devrait disparaître, et beaucoup d'artefacts sont visibles dans l'image prédite) en revanche notre algorithme parvient encore à fournir une estimation correcte de l'image d'arrivée, comme les données du tableau 2.B le prouvent quantitativement. Malgré la différence importante entre l'image de départ et d'arrivée, notre méthode est parvenue à fournir une bonne prédiction.

Ce test montre que notre algorithme permet des prédictions à partir d'images très différentes. Nous avons enfin testé un encodage d'images successives pour montrer que notre algorithme n'a pas besoin de beaucoup d'images de références pour la compression de séquences vidéo, ce qui implique une diminution du taux de compression. Chaque nouvelle image est prédite uniquement à partir de la prédiction de l'image précédente (c'est à dire sans correction de l'erreur de prédiction). La figure 6 montre l'évolution du *PSNR* des prédictions. Malgré une baisse progressive de ce *PSNR* (prévisible), il est toujours au-dessus de 30, même après 15 prédictions successives. L'image est donc correctement caractérisée par les vecteurs de déplacement et par leurs valeurs λ associées. Il n'est alors pas forcement nécessaire d'encoder l'erreur de prédiction (selon la qualité désirée) et le nombre d'images de référence nécessaires diminue, impliquant un gain de place significatif.

5 Conclusion

Nous avons présenté un algorithme permettant d'améliorer l'étape de compensation de mouvement d'une chaîne de

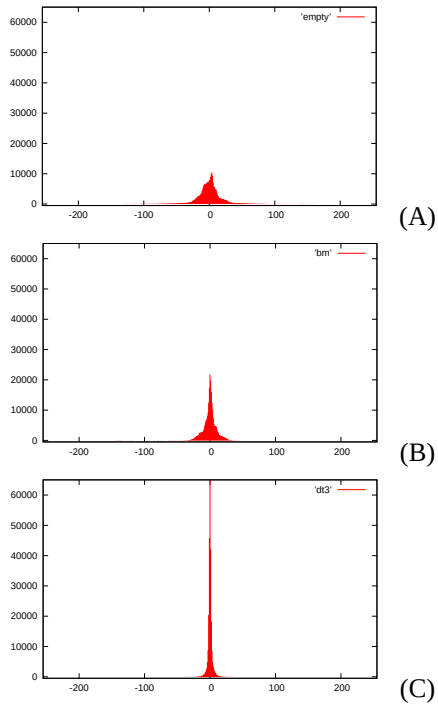


Figure 3 – *Histogrammes de la différence entre l'image d'arrivée et (A) : l'image initiale, (B) : l'image prédite par Block-Matching et (C) : l'image prédite par notre approche.*

compression de séquences vidéo. Cet algorithme repose sur l'utilisation des distances tangentes. Contrairement aux méthodes de *Block-Matching* classiques, celui-ci a l'avantage de prédire l'évolution temporelle des blocs de l'image, ainsi que l'évolution des pixels au sein de ces blocs.
Les différents résultats montrent l'efficacité de cet algorithme, qui permet d'obtenir une faible erreur de prédiction et donc d'améliorer le taux de compression et la qualité de l'image reconstruite, et ce même en situation difficile. Nous avons également montré que l'on peut diminuer le nombre d'images de référence au sein d'une séquence, le taux de compression en étant d'autant meilleur.
Une réflexion est à conduire sur le choix des transformations à modéliser. Il serait intéressant d'adapter les transformations à chaque bloc afin d'obtenir la meilleur prédiction. Il est aussi envisageable, en modélisant les translations (en X et Y), d'obtenir une estimation subpixelique du déplacement du bloc sans suréchantillonner l'image.
L'étude présentée ici est volontairement indépendante d'une chaîne de compression spécifique et peut être adaptée à de nombreux encodeurs. Toutefois nos recherches vont maintenant s'orienter sur la spécification d'une chaîne qui tire au maximum profit des avantages de notre méthode.

Références

[1] B. Furht, B. Furht, et J. Greenberg. *Motion estimation algorithms for video compression*. Kluwer Academic Publishers, 1996.

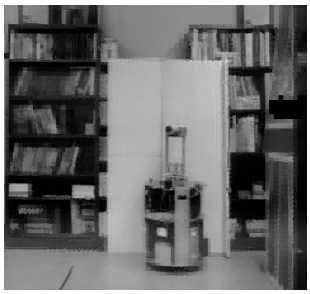
[2] L.M. Lopes Texeira et A.P. Alves. Block matching algorithms in mpeg video coding. Dans *13th International*

(A)

(B)

Figure 4 – *Prédiction de l'image de la figure 2.B à partir de l'image de la figure 2.A. (A) Par Block-Matching, l'image est bonne mais des artefacts apparaissent (notamment en haut du tonneau et autour du paravent). (B). Par notre algorithme, la qualité de l'image est excellente.*

(A)



(B)

Figure 5 – *Prédiction de l'image de la figure 2.C à partir de l'image de la figure 2.A. (A). Par Block-Matching, l'image est inutilisable. (B). Par notre algorithme, la qualité de l'image est encore acceptable (le tonneau a bien disparu).*

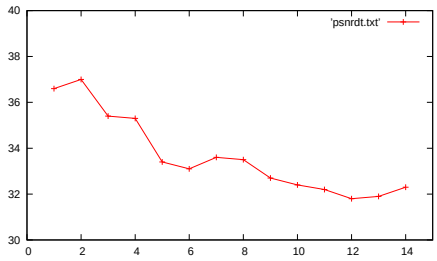


Figure 6 – *Évolution du PSNR des prédictions successives dans une séquence obtenues par notre algorithme. Malgré un baisse progressive, il reste au dessus de 30 après 15 images. L'intervalle entre deux images de référence peut donc être augmenté.*

Conference on Pattern Recognition (ICPR'96), pages 934–939, 1996.

[3] A. Gyaourova, C. Kamath, et S.-C. Cheung. Block matching for object tracking. Rapport technique, Novembre 2003.

[4] E. Chan et S. Panchanathan. Review of block matching based motion estimation algorithms for video compression. Dans *Canadian Conference on Electrical and Computer Engineering*, pages 151–154, 1993.

[5] J.-B. Xu, L.-M. Po, et C.-K. Cheung. Adaptive motion tracking block matching algorithms for video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(7) :1025–1029, 1999.

[6] N.S. Love et C. Kamath. An empirical study of block matching techniques for detection of moving objects. Rapport technique, Avril 2006.

[7] S. Martucci. Reversible compression of hdtv images using median adaptive prediction and arithmetic coding. Dans *Proc. IEEE International Symposium on Circuits and Systems*, pages 1310–1313, 1990.

[8] M. Weinberger, G. Seroussi, et G. Sapiro. Loco-i : a low complexity, context-based, lossless image compression algorithm. Dans *Proc. Data Compression Conference*, pages 140–149, 1996.

[9] J. Fabrizio et S. Dubuisson. Motion estimation using tangent distance. Dans *International Conference of Image Processing (ICIP'07)*, 2007.

[10] P. Simard, B. Victorri, Y. LeCun, et J. Denker. Tangent prop : a formalism for specifying selected invariances in adaptive networks. Dans *Advances in Neural Information Processing Systems 4 (NIPS*91)*, Denver, CO, 1992.

[11] P. Simard, Y. LeCun, J. Denker, et B. Victorri. Transformation invariance in pattern recognition, tangent distance and tangent propagation. Dans *Neural Networks : Tricks of the trade*, 1998.

[12] H. Schwenk et M. Milgram. Constraint tangent distance for on-line character recognition. Dans *International Conference on Pattern Recognition*, pages 515–519, 1996.

[13] H. Schwenk. *Amélioration de classifieurs neuronaux par incorporation de connaissances explicites : Applications à la reconnaissance de caractères manuscrits*. Thèse de doctorat, Université Pierre et Marie Curie - Paris VI, 1996.

[14] R. Mariani. A face location and recognition system based on tangent distance. Dans *Multimodal interface for human-machine communication*, pages 3–31, River Edge, NJ, USA, 2002. World Scientific Publishing Co., Inc.

[15] W. Macherey, D. Keysers, J. Dahmen, et H. Ney. Improving automatic speech recognition using tangent distance. Dans *Eurospeech 2001, 7th European Conference on Speech Communication and Technology*, volume III, pages 1825–1828, Aalborg, Denmark, 2001.

[16] Carnegie Mellon Image Database. http ://www.cs.cmu.edu/.