Minimisation hiérarchique pour le suivi des mouvements de la main

O. Ben Henia[1] M. Hariti[1] S. Bouakaz[1]

[1] LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information)

Université Claude Bernard, Lyon 1
43, Boulevard du 11 novembre 1918
69622 Villeurbanne Cedex

{obenheni, mhariti, sbouakaz}@liris.cnrs.fr

Résumé

Les approches de suivi des mouvements de la main à base de modèle 3D peuvent être classifiées en deux catégories. La première catégorie utilise des filtres stochastiques comme le filtre de Kalman ou le filtre particulaire. La deuxième se base sur des méthodes déterministes définissant une fonction de dissimilarité qui compare les gestes de la main avec ceux du modèle 3D. La minimisation de cette fonction assure le suivi des mouvements de la main. Deux principaux problèmes surviennent lors de la minimisation. Le premier problème est celui des minimas locaux et le deuxième est celui du temps de calcul nécessaire pour se rapprocher suffisamment de la solution recherchée. Pour faire face à ces deux problèmes nous proposons une nouvelle fonction de dissimilarité qui est plus robuste face aux minimas locaux que d'autres fonctions très connues comme la fonction de Chanfrein[1]. Nous proposons aussi un algorithme de minimisation hiérarchique qui simplifie et améliore le suivi des mouvements de la main en diminuant les temps de calcul et en améliorant la robustesse face aux minimas locaux.

Mots clefs

Suivi des mouvements de la main, minimisation, modèle 3D.

1 Introduction

Le suivi des gestes de la main est un domaine en pleine expansion. Cela est dû aux nombreuses applications qui en découlent comme par exemple la création d'une interface Homme-Machine (IHM) où selon le geste de la main une action spécifique est réalisée. Les gants de données, appelés aussi gants instrumentés, sont couramment utilisés comme périphérique d'entrée pour saisir et suivre le mouvement de la main grâce à des capteurs. Malgré leur efficacité à capturer les mouvements de la main, les gants de données sont très couteûx, très fragiles et leurs câbles de liaison constituent une entrave les rendant encombrants.

De nombreux travaux de recherche s'intéressent à d'autres alternatives et notamment à la vision artificielle pour la capture des mouvements de la main [2][3][4]. En effet, les caméras vidéo sont plus accessibles en termes de coût et de simplicité d'utilisation. Cependant, le suivi des mouvements de la main à base de caméras reste encore complexe à cause des problèmes d'occultation et du nombre élevé des degrés de liberté de la main.

Dans ce papier nous proposons une méthode orientée modèle 3D paramétrique pour suivre des mouvements de la main dans une séquence vidéo. Nous définissons une nouvelle fonction de dissimilarité qui compare les gestes de la main avec ceux du modèle 3D. Cette fonction est ensuite minimisée pour chaque image de la séquence vidéo pour obtenir les paramètres du modèle permettant de reproduire les mouvements de la main observés dans la séquence d'images. En raison du nombre élevé des degrés de liberté de la main (aux alentours de 26), beaucoup de paramètres sont à estimer pendant la phase de minimisation. Cela rend le processus de minimisation sensible aux minimas locaux et plus couteux en temps de calcul. C'est pourquoi nous proposons une minimisation hiérarchique de la fonction de dissimilarité. Ceci nous a permis de simplifier et d'améliorer le suivi des mouvements de la main en diminuant les temps de calculs et en améliorant la robustesse face aux minimas locaux.

Dans la section suivante nous présentons un bref état de l'art des approches de suivi des mouvements de la main. La section 3 décrit le modèle 3D ainsi que la fonction de dissimilarité qui compare la projection du modèle avec l'image de la main. La section 4 détaille l'algorithme de suivi. Avant de conclure nous présentons dans la section 5 les résultats expérimentaux obtenus à partir de séquences d'images synthétiques et réelles.

2 Etat de l'art

Les approches de suivi des mouvements de la main dans une séquence vidéo peuvent être décomposées en deux classes. La première classe utilise une base de gestes à partir de laquelle on cherche le geste correspondant à celui observé dans une image de la séquence vidéo. Ces approches utilisent en général des techniques de classification ou de

régression [5][6].

En raison de la grandeur de l'espace des gestes que peut prendre une main, il est difficile voire impossible d'obtenir une base contenant tous les gestes possibles d'une main. Ainsi, ces approches sont bien adaptées pour la reconnaissance d'un nombre fini de poses prédéfinies pour des applications temps réel. Dans ce cas, le temps de calcul est privilégié sur la précision du suivi. C'est le cas des interfaces homme-machine. Dans cette optique, un système de reconnaissance de gestes utilisant un classifieur est proposé dans [7].

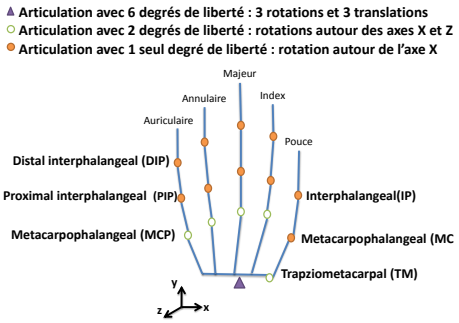
La deuxième classe regroupe des approches de suivi utilisant un modèle 3D paramétrique. Le problème du suivi est alors formalisé sous forme d'un problème d'estimation des paramètres du modèle 3D permettant de reproduire les gestes de la main observés dans une séquence vidéo. Les paramètres du modèle 3D peuvent être estimés en utilisant des méthodes stochastiques ou déterministes. Le premier type de méthodes utilise des filtres stochastiques comme le filtre de Kalman étendu utilisé dans [8] ou le filtre particulaire [9] [10]. Ce dernier donne de meilleurs résultats que le filtre de Kalman mais présente l'inconvénient d'être couteûx en temps de calcul. Outre les méthodes stochastiques, des méthodes déterministes ont aussi été utilisées pour réaliser le suivi des mouvements de la main. Dans ce cas, le problème de suivi est formalisé sous forme d'un problème de minimisation. En effet, une fonction de dissimilarité comparant les gestes de la main avec ceux du modèle 3D est définie. Cette dernière est minimisée afin d'estimer les paramètres du modèle 3D reproduisant les gestes de la main observés dans une séquence vidéo. Dans cette catégorie de méthodes, différentes fonctions de dissimilarité ont été proposées. Certaines se basent sur l'information de silhouette [11] tandis que d'autres définissent une distance au contour [1]. Delamarre et Faugeras [12] ont proposé un approche basée sur la stéréovision. Bray et al [13] ont défini une fonction qui utilise l'information contenue dans une carte de profondeur de la main. Cette carte de profondeur est obtenue grâce à des capteurs spécifiques.

Dans notre travail, nous définissons une nouvelle fonction de dissimilarité qui donne de meilleurs résultats que d'autres fonctions très connues comme la celle de Chanfrein [1] ou celle de la surface de non recouvrement utilisée dans [11]. Nous proposons par la suite de minimiser cette fonction en deux étapase en utilisant l'algorithme de Torczon[14]. La première étape donne les paramètres du modèle relatifs à la position et l'orientation de la paume de la main. La deuxième estime les angles d'articulations des doigts. Cette simplification nous a permis d'améliorer les temps de calculs et la robustesse face aux minimas locaux.

3 Modèle 3D et Fonction de Dissimilarité

3.1 Modèle 3D de la main

Le modèle 3D utilisé est un modèle paramétrique respectant la norme H-Anim. Ce modèle possède une partie cinématique et une partie apparence. Pour cette dernière nous utilisons des quadriques telles que des sphères, ellipsoïdes et cônes pour donner une forme au modèle 3D proche de celle de la main (Figure.1(b)).



(a)

(b)

Figure 1 – *(a) Représentation squelettique du modèle 3D montrant ces différentes articulations (b) Apparence du modèle 3D à base quadriques*

La partie cinématique est constituée d'une hiérarchie de transformations 3D (rotations, translations) permettant d'animer le modèle 3D. On peut énumérer 26 paramètres correspondants aux degrés de liberté de la main. Les six premiers paramètres du modèle modélisent des mouvements globaux de la main : rotations et translations de la paume de la main. Les 20 paramètres restants modélisent des mouvements locaux ou plus fins de la main : les articulations des doigts. En effet, chaque doigt peut être modélisé par 4 degrés de libertés : deux pour les articulations MCP et leur abduction et deux correspondants aux articulations PIP et DIP. Nous exploitons la dépendance entre les angles DIP et PIP pour réduire la partie cinématique de notre modèle à 22 degrés de liberté. La formule utilisée est comme suit : $\theta_{DIP} = 2/3\theta_{PIP}$.

En utilisant ce modèle 3D nous allons générer des projections qui seront comparées avec les images de la main. Cette comparaison est réalisée grâce la fonction de dissimilarité présentée dans la sous-section suivante.

3.2 Fonction de dissimilarité

Parmi les fonctions les plus connues comparant les images de la main avec les projections du modèle 3D on peut citer les fonctions qui estiment une distance entre deux contours : celui extrait de l'image de la main (Figure.2(e)) et celui de la projection du modèle 3D. C'est le cas de la fonction de Chanfrein. Celle-ci estime une distance entre deux contours en utilisant la distance de Chanfrein. En effet, à partir de deux ensembles de pixels A et B représentant les contours extraits de deux images I_a et I_b une valeur de dissimilarité d_C est calculée. La fonction de Chanfrein d_C estimant une distance entre deux contours A et B peut être exprimée selon la formule suivante :

$$d_C(A,B) = \frac{1}{|A|}\sum_{a_i \in A} \min_{b_j \in B} d(a_i, b_j) \qquad (1)$$

où d est une approximation de la distance euclidienne calculée grâce à l'algorithme de Chanfrein [1].

La fonction de Hausdorff est elle aussi très connue et peut être considérée comme une variante de la fonction de Chanfrein. En reprenant les mêmes notations utilisées pour définir la fonction de Chanfrein, la fonction de Hausdorff peut être formulée de cette manière :

$$d_H(A,B) = \max_{a_i \in A}\{\min_{b_j \in B} d(a_i, b_j)\} \qquad (2)$$

Ces fonctions qui se basent sur les contours sont très sensibles au bruit présent dans les images. Une autre alternative est proposée par Ouhadi et Horrain[11] en calculant la surface de non recouvrement(Figure.2(d)) correspondant à la partie non commune aux deux surfaces : la silhouette de la main H_s (Figure.2(a)) et la projection du modèle M_p (Figure.2(b)). Notons SNR une image de dimension $N_l \times N_c$ contenant la surface de non recouvrement (Figure.2(d)), où N_l et N_c représentent le nombre de lignes et de colonnes respectivement. Un pixel (i,j) de l'image SNR est défini par :

$$SNR_{ij} = \begin{cases} 1 & \text{si le pixel (i,j) appartient à la surface} \\ & ((H_s \cup M_s) - (H_s \cap M_s)) \\ 0 & \text{sinon} \end{cases}$$

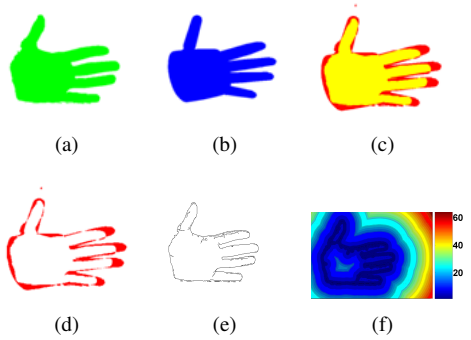


Figure 2 – *Différentes images utilisées pour le calcul de notre fonction de dissimilarité :(a) Silhouette de la main H (b) Projection du modèle M_p(c) Superposition de la silhouette de la main et de la projection du modèle (d) Surface de non recouvrement SNR (e) Contour de la main (f) Carte de distance D*

Pour rendre plus robsuste la fonction de non recouvrement, nous proposons d'ajouter une pondération à chaque pixel de la surface de non recouvrement SNR. Cette pondération est calculée à partir de la carte de distance D (Figure.2(f)) obtenue en appliquant l'algorithme de Chanfrein à l'image contenant le contour de la main(Figure.2(e)). Un élément D_{ij} de la carte D contient la distance d'un pixel (i,j) au contour de la main(Figure.2(e)). Ainsi, notre fonction de dissimilarité compare l'image de la main avec la projection du modèle associée aux paramètres R,T et θ, où R et T représentent le mouvement global de la paume de la main (3 rotations et 3 translations) et θ représente le mouvement local (angles d'articulation des doigts). Notre fonction de dissimilarité d_F est formalisée comme suit :

$$d_F(SNR, D) = \sum_{i=1,j=1}^{N_r,N_c} SNR_{ij} * D_{ij} \qquad (3)$$

4 Algorithme de Suivi

Le suivi des gestes de la main dans une séquence vidéo est réalisé en estimant les paramètres du modèle 3D permettant de reproduire le mouvement de la main observé dans la séquence vidéo. Cette estimation est réalisée en minimisant la fonction de dissimilarité pour chaque image de la séquence vidéo. Ceci permet de recaler la projection du modèle 3D sur la surface de la main pour chaque image et ainsi reproduire le mouvement observé dans la séquence vidéo.
Pour la première image de la séquence vidéo, nous supposons que les paramètres du modèle 3D sont proches de la solution recherchée. Pour le reste de la séquence vidéo, l'algorithme de minimisation est initialisé à partir des paramètres du modèle 3D de la main obtenus à l'image précédente. L'algorithme de minimisation utilisé est celui de Torczon[14] qui présente une amélioration de l'algorithme

du simplexe proposé par Nelder et Mead[15]. En effet, la méthode de Torczon[14] ne présente pas des problèmes de dégénérescence comme c'est le cas pour la méthode de descente du simplexe proposée par Nelder et Mead[15]. L'algorithme de Torczon[14] est un processus itératif explorant à chaque itération différentes directions pour en choisir celle qui minimise au mieux la fonction de dissimilarité. Une des particularités de cet algorithme est qu'il ne requiert pas la connaissance de la dérivée de la fonction à minimiser. Le deuxième argument justifiant notre choix de l'algorithme de Torczon[14] est lié au traitement de celui-ci qui explore différentes directions à chaque itération. Cette recherche multidirectionnelle peut être exécutée en parallèle afin d'améliorer les temps de calcul nécessaires pour atteindre la solution recherchée.
En raison de la grande dimensionnalité de l'espace de recherche, nous découpons l'algorithme de minimisation en deux étapes. La première étape estime les paramètres du modèle 3D relatifs au mouvement global de main : la translation et la rotation de la paume de la main. Ainsi, dans cette première étape, les paramètres qui représentent les angles des articulations des doigts sont fixes, tandis que ceux qui représentent la position et l'orientation de la main sont traités par l'algorithme de minimisation. Le processus est inversé lors de la deuxième étape, c'est-à-duire les paramètres d'orientation et de position sont tout d'abord fixés à ceux obtenus dans la première étape, et les angles des articulations des doigts sont ensuite estimés. Ce processus est résumé dans le schéma (Figure.3).

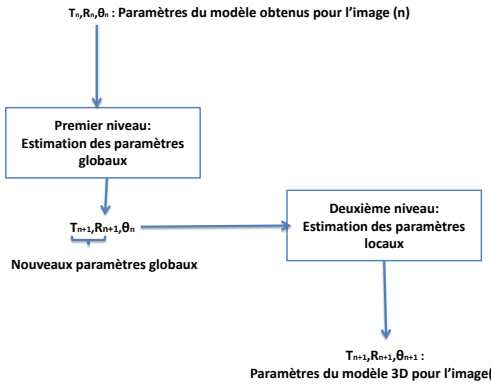


Figure 3 – *Processus d'estimation en deux étapes des paramètres du modèle 3D*

Outre la simplification du problème de minimisation, cette approche peut être justifiée par la variation lente du mouvement de la main entre deux images successives.

5 Résultats Expérimentaux

Les performances de notre algorithme de suivi des mouvements de la main sont évaluées sur des séquences d'images vidéo synthétiques et réelles. Nous évaluons notre fonction de dissimilarité en la comparant avec d'autres fonctions de comme celle de Chanfrein [1], celle de Hausdorff [16] ainsi que celle de non recouvrement [11].Pour cela, une séquence vidéo composée d'une centaine d'images de synthèse de dimension 320x240 est acquise à partir du mo-

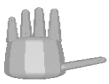
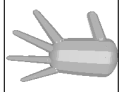
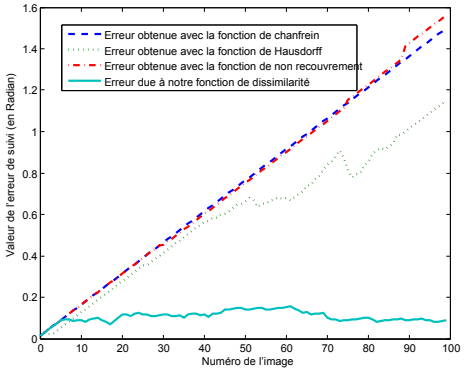
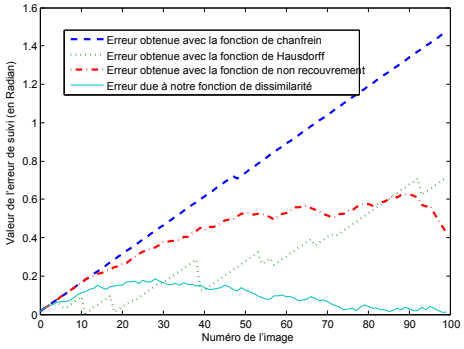
	Algorithme de suivi	Image 1	Image 50	Image 100
Poses à retrouver				
d_C	une étape			
	deux étapes			
d_H	une étape			
	deux étapes			
d_{SNR}	une étape			
	deux étapes			
d_F	une étape			
	deux étapes			

Tableau 1 – *Résultats du suivi obtenus par l'algorithme à une étape et celui à deux étapes en utilisant différentes fonctions de dissimilarité : fonction de Chanfrein(d_C), fonction de Hausdorff (d_H), fonction de non recouvrement(d_{SNR})et notre fonction proposée(d_F).*

dèle 3D de la main (Tableau 1). Pour obtenir cette séquence d'images, on fait varier trois (respectivement quatre) paramètres relatifs au mouvement global (respectivement local). Les paramètres du mouvement global sont la translation selon les axes X et Y ainsi que la rotation autour de l'axe des Z (Fig :1(a)). Les paramètres locaux sont les articulations MCP métacarpophalangienne (Fig :1(a)) et les abductions des doigts de la main excepté le pouce. Les résultats du suivi dans la vidéo de synthèse sont présentés dans le tableau 1.



(a)



(b)

Figure 4 – *Erreur de suivi de la rotation de la main autour de l'axe Z(a) Erreurr obtetnue en utilisant l'algorithme de minimisation à une étape(b) Erreur obtenue par l'algorithme de minimisation hiérarchique*

Dans le même tableau, nous pouvons voir que notre fonction de dissimilarité d_F fournit les meilleurs résultats comparés avec ceux des autres fonctions de dissimilarité : la fonction de Chanfrein d_C[1], la fonction de Hausdorff d_H [16] ou encore la fonction de non recouvrement (d_{SNR})[11].
Pour quantifier l'erreur de suivi associée à chaque fonction de dissimilarité, on calcule une différence entre les résultats du suivi obtenus et les valeurs exactes recherchées. L'erreur de suivi est alors tracée sous forme d'une courbe (Figure.4). Nous traçons seulement la courbe représentant l'erreur relative au suivi de la rotation autour de l'axe Z. Nous observons dans la figure4 que notre fonction de dissimilarité est plus efficace que les autres fonctions de dissimilarité. En effet, l'erreur de suivi est de 0,09 radian pour notre fonction de dissimilarité, alors que celle-ci peut être supérieure à 1 radian avec les autres fonctions, notamment pour la fonction Chanfrein. La même figure montre également que la minimisation hiérarchique est plus robuste qu'une minimisation en une étape. Plus précisément, dans le cas de la fonction de non recouvrement, la minimisation hiérarchique améliore considérablement les performances du suivi en diminuant l'erreur de suivi d'un rapport de 1/2 (figure4). Nos observations concernant l'erreur de suivi du mouvement de rotation autour de l'axe Z restent valables pour l'estimation des autres paramètres du modèle 3D. Outre la robustesse, la minimisation hiérarchique est plus rapide en temps de calcul qu'une minimisation en une étape. En effet, pour la séquence d'images de synthèse, la cadence de traitement est d'environ 8 images par seconde pour une minimisation en une étape alors que la minimisation hiérarchique a une cadence de 11 images par seconde. Ces résultats ont été obtenus en utilisant un processeur à 2.2GHZ. Les projections du modèle 3D sont calculées en utilisant la librairie graphique d'OpenGL.
Nous évaluons également notre algorithme de suivi sur une séquence d'images réelles (Figure.5).Dans la figure5 la ligne du haut montre des images de la vidéo traitée. La deuxième ligne montre les résultats du suivi obtenus par l'algorithme de minimisation à une étape. La dernière ligne montre les résultats du suivi obtenus par notre algorithme de minimisation hiérarchique. La figure 5 montre aussi la robustesse du suivi en utilisant une minimisation hiérarchique. En effet, nous pouvons souligner que la minimisation hiérarchique est plus efficace pour suivre les mouvements complexes des doigts de la main comme on peut le voir notamment dans l'image numéro 400, où le suivi des doigts se perd dans le cas de la minimisation en une étape.

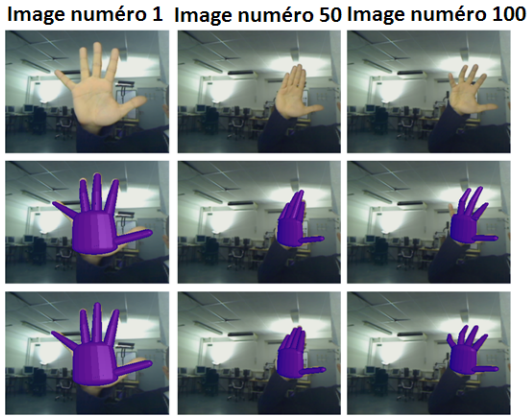


Figure 5 – *Résultats du suivi de la main dans une séquence d'images réelles*

6 Conclusion et Travaux Futurs

Dans cet article, nous avons proposé une méthode de suivi de suivi des mouvements de la main à partir d'une caméra,

sans marqueur et en utilisant un modèle 3D paramétrique de la main. Une nouvelle fonction de dissimilarité comparant des gestes de la main avec ceux du modèle 3D est proposée. Cette fonction est minimisée pour estimer les paramètres du modèle 3D reproduisant le mouvement de la main. Le grand nombre des degrés de liberté (environ 26) qui doivent être estimées rend la minimisation sensible aux minimas locaux et augmente le temps de calcul nécessaire pour atteindre la solution recherchée.
Une minimisation hiérarchique en deux étapes de la fonction de dissimilarité a permis de simplifier le problème de minimisation. La première étape de notre minimisation hiérarchique estime les degrés de libertés globaux de la main : position et orientation de la paume. La deuxième étape estime les degrés de libertés locaux de la main, c'est-à-dire les angles des articulations des doigts. D'après nos résultats expérimentaux, l'algorithme de minimisation hiérarchique est plsu robuste face aux minimas locaux qu'un algorithme classique de minimisation en une étape.
L'algorithme que l'on propose améliore également la rapidité du suivi des mouvements de la main. Dans de cadre de nos travaux de recherche, nous avons utilisé une seule caméra et il reste difficile de traiter, dans le cas mono-caméra, des mouvements complexes tels que des mouvements de torsion de la main. Nous sommes effectivement très vite confrontés au problème de l'auto-occultation. L'utilisation de plusieurs caméras ou encore d'autres technologies comme les caméras à temps de vol (exemple la swissranger [1]) peut être une solution pour suivre des mouvements plus complexes de la main. L'utilisation de plusieurs points de vue dans le cas multi-caméras ou celle de l'information 3D dans le cas des caméras à temps de vol peut résoudre certaines ambigüités. Des études en cours d'approfondissement tentent d'améliorer les temps de calcul en transférant une partie des traitements sur des unités de traitement graphiques appelées GPUs.

Références

[1] G. Borgefors. Hierarchical chamfer matching : A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6) :849–865, 1988.

[2] Ying Wu et Thomas S. Huang. Vision-based gesture recognition : A review. Dans *GW '99 : Proceedings of the International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction*, pages 103–115, London, UK, 1999. Springer-Verlag.

[3] Martin Tosas. *Visual Articulated hand tracking for Interactive Surfaces*. Thèse de doctorat, University of Nothingham, 2006.

[4] Bjorn Dietmar Rafael Stenger. *Model-Based Hand Tracking Using A Hierarchical Bayesian Filter*. Thèse de doctorat, University of Cambridge, 2004.

1. http ://www.mesa-imaging.ch/

[5] Rómer Rosales, Vassilis Athitsos, Leonid Sigal, et Stan Sclaroff. 3d hand pose reconstruction using specialized mappings. Dans *ICCV*, pages 378–385, 2001.

[6] Nobutaka Shimada, Kousuke Kimura, et Yoshiaki Shirai. Real-time 3-d hand posture estimation based on 2-d appearance retrieval using monocular camera. Dans *RATFG-RTS '01 : Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'01)*, page 23, Washington, DC, USA, 2001. IEEE Computer Society.

[7] Tsukasa Ike, Nobuhisa Kishikawa, et Björn Stenger. A real-time hand gesture interface implemented on a multi-core processor. Dans *MVA*, pages 9–12, 2007.

[8] B. Stenger, P. R. S. Mendonca, et R. Cipolla. Model-based 3d tracking of an articulated hand. Dans *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–310–II–315 vol.2, 2001.

[9] Michael Isard et Andrew Blake. Condensation conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29 :5–28, 1998.

[10] Makoto Kato et Gang Xu. Occlusion-free hand motion tracking by multiple cameras and particle filtering with prediction. *IJCSNS International Journal of Computer Science and Network Security*, 6(10) :58–65, 2006.

[11] Hocine Ouhaddi et Patrick Horain. 3d hand gesture tracking by model registration. Dans *Proc.IWSNHC3DI99*, pages 70–73, 1999.

[12] Quentin Delamarre et Olivier Faugeras. Finding pose of hand in video images : a stereo-based approach. Dans *In IEEE Proc. of the third International Conference on Automatic Face and Gesture Recognition*, pages 585–590. IEEE Computer Society, 1998.

[13] Matthieu Bray, Esther Koller-Meier, Nicol N. Schraudolph, et Luc Van Gool. Stochastic meta-descent for tracking articulated structures. Dans *CVPRW '04 : Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 1*, page 7, Washington, DC, USA, 2004. IEEE Computer Society.

[14] J. E. Dennis, Jr., et Virginia Torczon. Direct search methods on parallel machines. *SIAM Journal on Optimization*, 1 :448–474, 1991.

[15] J. A. Nelder et R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4) :308–313, January 1965.

[16] Daniel P. Huttenlocher, Gregory A. Klanderman, Gregory A. Kl, et William J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15 :850–863, 1993.