



HAL
open science

Nouvelle méthode de suivi des mouvements la main utilisant des nuages de points 3D

Ouissem Ben-Hania, Saïda Bouakaz

► **To cite this version:**

Ouissem Ben-Hania, Saïda Bouakaz. Nouvelle méthode de suivi des mouvements la main utilisant des nuages de points 3D. ORASIS - Congrès des jeunes chercheurs en vision par ordinateur, INRIA Grenoble Rhône-Alpes, Jun 2011, Praz-sur-Arly, France. inria-00595487

HAL Id: inria-00595487

<https://inria.hal.science/inria-00595487v1>

Submitted on 24 May 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.

Nouvelle méthode de suivi des mouvements la main utilisant des nuages de points 3D.

Ouissem Ben-Henia

Saida Bouakaz

LIRIS CNRS UMR 5205

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

ouisseben-henia@liris.cnrs.fr ; saida.bouakaz@liris.cnrs.fr

Résumé

Plusieurs méthodes de suivi des mouvements de la main à partir d'une séquence vidéo s'appuient sur un modèle 3D paramétrique de la main. Ces méthodes définissent une fonction de comparaison, appelée aussi fonction de coût ou encore fonction de dissimilarité pour comparer les poses de la main observées dans une séquence vidéo avec celles du modèle 3D. Le suivi des mouvements de la main est alors formulé comme un problème de recherche des paramètres du modèle 3D permettant de minimiser la fonction de coût. La plupart des fonctions de coût exploitent des informations 2D comme les informations de contours et de silhouettes contenues dans les images de la séquence vidéo. Ces fonctions donnent des résultats pour suivre des mouvements simples de la main comme le suivi de la position dans l'espace. Malheureusement ces fonctions ne sont plus efficaces pour faire face à des mouvements complexes. Dans ce papier nous présentons une méthode de suivi s'appuyant sur une fonction qui compare des données 3D. Les gestes de la main sont représentés sous forme de nuages de points 3D. Ces derniers sont comparés à d'autres nuages de points générés à partir d'un modèle 3D de la main. La fonction de comparaison est minimisée afin de trouver les paramètres du modèle 3D reproduisant les mouvements observés dans la séquence vidéo. Des résultats obtenus à partir de données synthétiques montrent que l'utilisation de l'information 3D permet un meilleur suivi des mouvements complexes de la main.

Mots Clef

Suivi de mouvements, Minimisation, Modèle 3D

Abstract

3D model-based methods for hand tracking use generally a 3D parametric hand model. These methods define a comparison function, also called cost function or dissimilarity function to compare the observed hand poses in video sequence with those of the 3D model. The tracking

of the hand motion is formulated as a minimization problem where the cost function is minimized to provide the model parameters which reproduce the hand motion. Many cost functions exploit 2D information such as edges and silhouettes contained in the frames of the video sequence. These functions allow tracking simple hand motion. Unfortunately these kinds of functions can't deal with the tracking of complex hand motion. This paper presents a new depth-based function to track complex hand motion such as opening and closing hand. Our proposed function compares 3D point clouds. Each hand point cloud is compared with several clouds of points which correspond to different model poses in order to obtain the model pose that is close to the hand one. The proposed function is based on 3D distance transform and well adapted to our case. Experimental results from synthetic data show the potential of the presented method.

Keywords

Hand tracking, Minimization, 3D model

1 Introduction

Dans cet article nous nous intéressons aux mouvements complexes de la main dans le but de développer des approches de suivi et de reproduction des mouvements de la main observés dans des séquences vidéo. De nombreux travaux de recherche concernant le suivi et la reproduction des mouvements de la main ont été proposés durant ces dernières années [1][2][3]. Ceci est dû aux nombreuses applications qui en découlent comme la création d'interfaces homme-machine(IHM) utilisées dans le domaine des jeux vidéo et de l'animation 3D. La plupart des méthodes que l'on trouve dans la littérature peuvent se classer selon deux catégories : Les méthodes utilisant une base de gestes et celles qui s'appuient sur un modèle 3D de la main.

Dans la première catégorie une base contenant des gestes de la main est utilisée. Le suivi consiste alors à rechercher les gestes de la base correspondant à ceux observés

dans une séquence vidéo. Ces approches utilisent en général des techniques de classification ou de régression [4][5]. En raison de la grandeur de l'espace des gestes possibles de la main, il devient très difficile voire même impossible de construire une base dense contenant tous les gestes de la main. Ainsi ces approches ont été utilisées au départ pour reconnaître quelques gestes de la main dans le but de créer des applications temps réel comme les interfaces homme-machine [6]. Aujourd'hui des travaux utilisant une base contenant plusieurs milliers de gestes ont été proposés [2] [3]. Le problème devient alors un problème de parcours de ces bases de manière rapide et efficace pour reconnaître les gestes observés dans une séquence d'images en entrée. Dans [3], Okada et Stenger représentent la base des gestes sous forme d'un arbre permettant un parcours arborescent de la base. Wan et Popovic [2] ont représenté les éléments de leur base de données sous une forme la plus réduite possible. En effet, chaque geste a été représenté sous forme d'un vecteur binaire. La comparaison des gestes revient alors comparer les vecteurs binaires ce qui a réduit les temps de calcul nécessaire à la comparaison des gestes et ainsi accélérer le parcours de la base de données. Malgré le nombre important des gestes contenus dans la base de données le suivi des mouvements de la main n'est pas très fluide.

Dans la deuxième catégorie des méthodes de suivi des mouvements de la main, un modèle 3D paramétrique de la main est souvent utilisé [7] [8] [9] [1]. Le problème de suivi est alors formulé comme un problème de recherche des paramètres du modèle 3D reproduisant le mouvement de la main observé dans une séquence vidéo. Afin de trouver les bons paramètres du modèle 3D deux types de méthodes ont été utilisées : Les méthodes stochastiques et les méthodes déterministes. Parmi les méthodes stochastiques nous pouvons citer le filtre de Kalman étendu utilisé dans [10] et le filtre particulaire utilisé dans [11] [12]. Ce dernier donne de meilleurs résultats que le filtre de Kalman mais présente l'inconvénient d'être très coûteux en temps de calcul. Outre ces méthodes stochastiques, les méthodes déterministes définissent le problème de suivi comme un problème de minimisation de fonction. En effet la fonction qui compare les gestes de la main avec ceux du modèle 3D est minimisée. La solution de cette minimisation correspond aux paramètres du modèle 3D reproduisant le mouvement de la main observé dans une séquence d'images. Dans cette catégorie de méthodes, la plupart des fonctions proposées dans la littérature utilisent des informations 2D comme la fonction de non recouvrement qui se base sur l'information de silhouette, ou la fonction de distance au contour. Dans [1] une fonction combinant distance au contour et surface de non recouvrement a été proposée. La Gorce et al proposent dans [7] une fonction utilisant la texture. Une fonction de couleur a été proposée dans [2]. Les fonctions de comparaison utilisant des données 3D sont peu nombreuses dans la littérature. Delamarre et Faugeras ont proposé une approche basée sur la stéréovision [13]. Bray et al [14] 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 cet article nous proposons une nouvelle méthode s'appuyant sur un modèle 3D qui utilise des nuages de points 3D. En effet chaque geste de la main est représenté sous forme d'un nuage de points 3D. Ces nuages de points 3D représentant la main sont comparés avec des nuages de points obtenus à partir du modèle 3D. Afin de comparer les nuages de points 3D nous utilisons une fonction de comparaison se basant sur les transformés de distance 3D. Nous minimisons cette fonction de comparaison en utilisant l'algorithme de Torczon [15] qui est une amélioration de l'algorithme de la descente du simple proposé par Nelder et Mead [16]. La minimisation de la fonction de comparaison pour chaque image d'une séquence vidéo donne les paramètres du modèle 3D reproduisant les gestes de la main observés dans la séquence d'images. La section suivante présente le modèle 3D utilisé pour suivre et reproduire les mouvements de la main. Dans la section 3 nous présentons la fonction utilisée pour comparer les nuages de points 3D. La section 4 détaille l'algorithme de suivi proposé. L'utilisation de données 3D donne de meilleurs résultats que l'utilisation de données 2D comme le montre la section 5 où nous présentons des résultats expérimentaux obtenus à partir d'une séquence d'images synthétiques. Nous terminons cet article par une conclusion dans laquelle nous discutons sur l'amélioration de la méthode proposée.

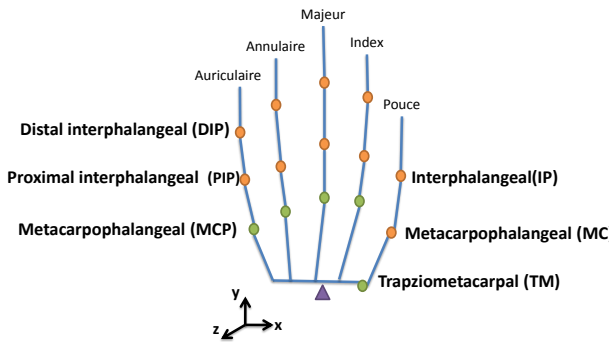
2 Modèle 3D de la main

La structure de la main est une structure articulée et très complexe. Plusieurs modèles ont été proposés dans la littérature. Dans [17] un modèle 3D formé d'un maillage complètement déformable a été proposé. Ce modèle ne peut pas reproduire avec précision tous les mouvements de la main de manière réaliste. En effet, ce modèle n'est pas basé sur un squelette rigide ; les doigts du modèle peuvent être déformés et réduits pour se recaler sur les images de la main. Un autre type de modèle basé sur un squelette est souvent utilisé [1] [2]. Ce type de modèle possède des paramètres qui sont généralement définis comme représentant des transformations hiérarchiques qui correspondent aux degrés de liberté de la main : la position et l'orientation de la paume de la main dans l'espace ainsi que les angles d'articulations des doigts (Figure.1(a)). On peut animer ce type de modèle 3D de la main en modifiant les valeurs ces paramètres. Grâce à ce type de modèle il est possible d'estimer non seulement la position et l'orientation de la main mais aussi les angles d'articulation des doigts.

Dans notre cas, nous utilisons un modèle paramétrique de la main qui est conforme à la norme H-Anim¹. Le modèle H-Anim est souvent utilisé dans le domaine de l'animation 3D. Nous pouvons mettre en évidence sa particularité

1. Humanoid-Animation (H-Anim) est un standard approuvé par la norme internationale ISO. Ce type de modèle est très utilisé pour la description et l'animation de personnage 3D. site internet : www.h-anim.org

- ▲ Articulation avec 6 degrés de liberté : 3 rotations et 3 translations
- Articulation avec 2 degrés de liberté : rotations autour des axes X et Z
- Articulation avec 1 seul degré de liberté : rotation autour de l'axe X



(a)



(b)

FIGURE 1 – (a) Représentation squelettique du modèle 3D
(b) Apparence du modèle 3D

de séparer la partie cinématique (mouvement) de la partie apparence. Ce modèle est constitué d'une hiérarchie des transformations 3D (rotations, translations) permettant un contrôle facile de son animation en modifiant seulement les valeurs des transformations (les paramètres).

En ce qui concerne l'apparence, des objets appelés segments sont placés dans cette représentation hiérarchique pour fournir la forme de ce modèle (figure 1(b)). Pour modifier l'apparence de ce modèle, nous modifions les objets représentant l'apparence du modèle. Dans notre cas nous utilisons des quadriques qui sont des sphères des ellipsoïdes et des cônes pour modéliser l'apparence du modèle 3D (Figure.1(b)). La paume de la main est modélisée par une ellipsoïde tronquée. Chaque doigt est composé de trois cônes tronqués : un pour chaque phalange. Des demi-sphères ont été utilisées pour la clôture les extrémités de chaque cône. Le principal avantage de l'utilisation de ce genre de structures géométriques et la simplicité de modifier la taille des quadriques et ainsi adapter le modèle 3D à n'importe quelle main que l'on veut suivre.

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 l'articulation MCP et son abduction et deux correspondants aux articulations PIP et DIP (Figure1(a)). 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 nuages de points 3D qui seront comparés avec ceux qui représentent les gestes de la main. Cette comparaison est réalisée grâce la fonction de dissimilarité présentée dans la section suivante.

3 Fonction de dissimilarité

Nous proposons d'évaluer une distance entre deux nuages de points qui représentent respectivement la main et son modèle 3D sous-jacent. Deux fonctions bien connues sont souvent utilisés pour calculer la distance à partir d'un ensemble de points N à un autre M . La première est donnée par la formule suivante :

$$d_1(N, M) = \frac{1}{|N|} \sum_{n_i \in N} \min_{m_j \in M} d(n_i, m_j) \quad (1)$$

où $|N|$ est le cardinal de l'ensemble de point N , et $d(n_i, m_j)$ la distance Euclidienne entre n_i et m_j .

La deuxième fonction très connue est la transformé de distance de Hausdorff. Notons cette fonction d_2 . Cette fonction peut se formuler par l'équation suivante :

$$d_2(N, M) = \max_{n_i \in N} \{ \min_{m_j \in M} d(n_i, m_j) \} \quad (2)$$

où $d(n_i, m_j)$ est la même distance Euclidienne définie dans Eq.1.

Le calcul de distance par le biais de cette classe de fonctions est effectué d'une manière non-linéaire. En effet, le nuage de points M est balayé autant de fois que de points dans N . Dans notre cas d'étude, ces fonctions sont très coûteuses en temps de calcul. En effet pour chaque nuage de points de la main plusieurs distances sont calculées pour trouver le nuage de points du modèle 3D se rapprochant au mieux de celui de la main. Dans cet article nous proposons une fonction bien adaptée à notre cas d'étude. Cette fonction est basée sur les transformés de distance 3D. Elle comprend deux étapes. La première calcule un volume de voxels à partir du nuage de points N , où chaque voxel contient la distance du voxel au nuage de points N . Le volume cubique est utilisé dans la deuxième étape pour évaluer la distance entre le nuage de points N et un autre M . L'algorithme développé pour calculer le volume de voxels est celui proposé par Meijster et al dans [18], où il a été appliqué pour calculer des cartes de distances 2D à partir d'images 2D. Cet algorithme peut être étendu à un espace

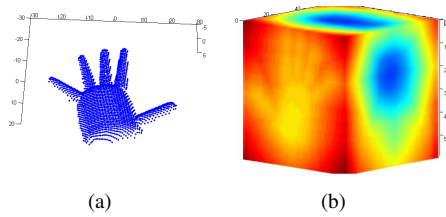


FIGURE 2 – (a) Nuage de point N (b) Volume de voxels CV obtenu à partir du nuage N

nD (n dimensions). Nous utilisons cet algorithme dans le cas 3D pour calculer un volume cubique CV (Figure 2(b)) du nuage de points N (Figure 2(a)). Dans la figure 2(b) la couleur correspond à une valeur de distance. Plus de détails peuvent être trouvés dans [18].

Une fois le volume 3D (CV) obtenu, nous pouvons estimer la distance entre le nuage N et un autre M par la formule suivante :

$$F(N, M) = \frac{1}{|M|} \sum_{m_j \in M} CV[m_{j_x}][m_{j_y}][m_{j_z}] \quad (3)$$

where m_{j_x} , m_{j_y} et m_{j_z} sont les coordonnées du point m_j .

4 Algorithme de suivi

Le suivi des gestes de la main dans une séquence vidéo est réalisé en recherchant les paramètres du modèle 3D de la main reproduisant le mouvement de la main comme le résume la figure 3. Pour y parvenir nous minimisons notre fonction de dissimilarité pour chaque image de la séquence vidéo. Cette minimisation fournit les paramètres du modèle qui permettent de rapprocher le geste du modèle 3D de celui de la main. Pour la première image de la séquence vidéo, nous supposons que les paramètres du modèle sont proches de la solution recherchée. Pour le reste de la séquence vidéo, le processus de minimisation est initialisé pour chaque image à partir des paramètres obtenus à l'image précédente. Nous utilisons un algorithme de minimisation itératif proposé par Torczon [15] qui explore à chaque itération différentes directions et garde celle qui minimise au mieux la fonction de dissimilarité. L'algorithme de Torczon est une amélioration de l'algorithme descente du simple de Nelder et Mead [16].

Le choix de l'algorithme de minimisation s'est fait pour deux raisons. La première raison est que l'algorithme de Torczon ne nécessite pas la connaissance de la dérivée de la fonction à minimiser. Il est d'ailleurs difficile de calculer la dérivée de notre fonction de dissimilarité. La seconde raison est que cet algorithme peut être parallélisé en explorant à chaque itération les différentes directions de manière parallèle.

5 Résultats Expérimentaux

Les performances obtenues par notre méthode sont évaluées pour le suivi des mouvements de la main apparais-

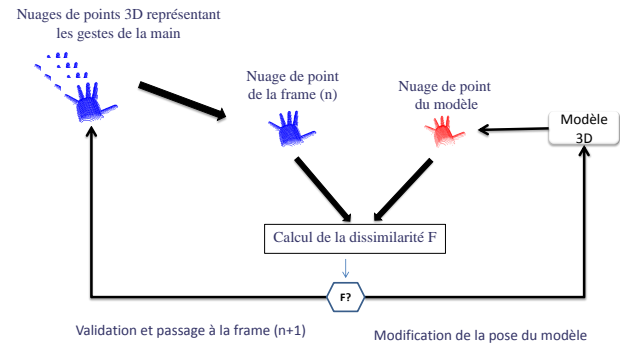


FIGURE 3 – Processus de suivi

sant dans des séquences d'images synthétiques. Une séquence vidéo de 100 images de résolution 320x240 sont acquises (Tableau 1). Pour obtenir cette séquence d'images, nous avons fait varier les paramètres de chaque doigt, sauf le pouce et l'index. Grâce à ce test de référence, nous pouvons souligner l'importance de l'information de profondeur pour faire face aux mouvements complexes de la main. Notre fonction proposée compare les nuages de points générés à partir des cartes de profondeur obtenues au moyen de la librairie graphique OpenGL.

TABLE 1 – Résultats de suivi : La première ligne représente les nuages de points 3D correspondant la vérité terrain. La deuxième ligne montre les résultats de suivi utilisant une fonction 2D s'appuyant sur l'information de silhouette. La troisième ligne montre les résultats obtenus en utilisant notre fonction 3D.

Frame 1	Frame 50	Frame 100

Nous comparons les résultats obtenus par notre fonction avec ceux obtenus par une fonction 2D proposée récemment dans [1]. Cette fonction utilise les silhouettes pour calculer une surface de non recouvrement. Les résultats ob-

servés dans le tableau 1 montrent que notre fonction est mieux adaptée pour suivre des mouvements complexes de la main comme la fermeture ou l'ouverture de la main.

En effet, le suivi de mouvement des doigts est perdu au bout de 50 images de la vidéo (Tableau 1). Pour estimer l'erreur de l'algorithme de suivi, nous calculons la différence entre le résultat du suivi et la vérité terrain. Dans la figure 4 nous représentons seulement l'erreur du suivi liée à l'articulation interphalange PIP (Figure 1(a)) du majeur. Les remarques faites pour cette articulation peuvent être généralisées aux autres articulations des autres doigts. Dans la figure 4 on peut voir que grâce à notre fonction, l'erreur est très faible si on la compare avec une autre fonction utilisant des information 2D(silhouette) [1]. En effet, la valeur de l'erreur moyenne du suivi de 0,5 radian avec la fonction 2D alors qu'elle est de 0,1 radian avec notre fonction 3D(Figure 4).

En terme de temps de calcul, nous avons comparé les résultats obtenus avec notre fonction 3D et ceux obtenus avec des fonctions classiques de comparaison de nuages de points 3D. Les résultats obtenus avec notre fonction prennent un temps approximatif de 3 secondes par image. L'utilisation de fonctions classiques comme la fonction de Hausdorff (voir Eq 2) prend un temps de calcul qui dépasse les 300 secondes par image.

Nous pouvons souligner le fait que le cardinal moyen des nuages de points utilisés dans notre test est d'environ 3000 points. Nous pouvons voir que notre fonction proposée est bien adaptée pour le suivi 3D de la main même avec des nuages de points assez volumineux. Le traitement est effectué en utilisant un PC Intel Centrino-processeur 2 GHz.

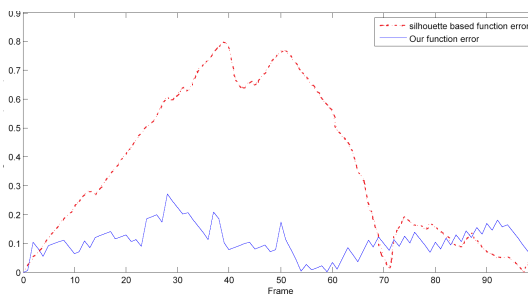


FIGURE 4 – Tracking error of the PIP middle finger

6 conclusion

Dans cet article nous proposons une méthode de suivi des mouvements de la main s'appuyant sur un modèle paramétrique 3D de la main. Les gestes de la main sont comparés avec ceux du modèle 3D afin de réaliser le suivi 3D de la main. Les gestes sont représentés sous forme de nuages de points 3D. Une fonction 3D adaptée à notre cas d'étude est proposée pour comparer les nuages de points 3D. L'utilisation de cette fonction 3D donne de meilleurs résultats

comparé avec des fonctions utilisant seulement des informations 2D comme la silhouette et les contours [1]. Nous comparons les temps calcul obtenus grâce à notre fonction avec ceux obtenus par la fonction de Hausdorff qui est une fonction classique qui estime la distance entre deux nuages de points. L'amélioration des temps de calcul grâce à notre fonction est alors de cent pour cent.

Nous prévoyons d'étendre notre étude expérimentale à des tests sur des données réelles. Dans ce but, plusieurs méthodes pour acquérir des données 3D doivent être testées. Parmi ces méthodes il y'a la stéréovision qui peut être envisagée pour acquérir des nuages de points 3D représentant la main. Cependant cette méthode nécessite l'utilisation de deux caméras et de faire un calibrage ce qui peut être considéré comme une contrainte. Une autre alternative consiste à utiliser une nouvelle génération de caméras actives appelées caméras 3D ou encore à temps de vol. Ces caméras arrivent à acquérir des nuages de points 3D en temps réel et à partir d'un seul point de vue. Cependant ces caméras ne sont pas connues pour être très précises. Une étude comparative de ces différentes méthodes doit donc être réalisée pour trouver la meilleure méthode permettant d'étendre nos travaux à des données réelles.

Références

- [1] Ouissem Ben Henia, Mohamed Hariti, and Saida Bouakaz. A two-step minimization algorithm for model-based hand tracking. In *WSCG*, 2010.
- [2] Robert Y. Wang and Jovan Popović. Real-time hand-tracking with a color glove. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, pages 1–8, New York, NY, USA, 2009. ACM.
- [3] Ryuzo Okada and Björn Stenger. A single camera motion capture system for human-computer interaction. *IEICE - Trans. Inf. Syst.*, E91-D(7) :1855–1862, 2008.
- [4] Rómer Rosales, Vassilis Athitsos, Leonid Sigal, and Stan Sclaroff. 3d hand pose reconstruction using specialized mappings. In *ICCV*, pages 378–385, 2001.
- [5] Nobutaka Shimada, Kousuke Kimura, and Yoshiaki Shirai. Real-time 3-d hand posture estimation based on 2-d appearance retrieval using monocular camera. In *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.
- [6] Tsukasa Ike, Nobuhisa Kishikawa, and Björn Stenger. A real-time hand gesture interface implemented on a multi-core processor. In *MVA*, pages 9–12, 2007.
- [7] Martin de La Gorce, Nikos Paragios, and David J. Fleet. Model-based hand tracking with texture, shading and self-occlusions. In *CVPR*, 2008.
- [8] Björn Stenger, Arasanathan Thayananthan, Philip H. S. Torr, and Roberto Cipolla. Model-based

hand tracking using a hierarchical bayesian filter. *IEEE Trans. Pattern Analysis and Machine Intelligence(PAMI)*, 28(9) :1372–1384, 2006.

- [9] Chutisant Kerdvibulvech and Hideo Saito. Model-based hand tracking by chamfer distance and adaptive color learning using particle filter. *J. Image Video Process.*, 2009 :2–2, 2009.
- [10] B. Stenger, P. R. S. Mendonca, and R. Cipolla. Model-based 3d tracking of an articulated hand. In *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.
- [11] Michael Isard and Andrew Blake. Condensation conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29 :5–28, 1998.
- [12] Makoto Kato and 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.
- [13] Quentin Delamarre and Olivier Faugeras. Finding pose of hand in video images : a stereo-based approach. In *In IEEE Proc. of the third International Conference on Automatic Face and Gesture Recognition*, pages 585–590. IEEE Computer Society, 1998.
- [14] M. Bray, E. Koller-Meier, N. N. Schraudolph, and L. Van Gool. Fast stochastic optimization for articulated structure tracking. *Image Vision Comput.*, 25(3) :352–364, 2007.
- [15] J. E. Dennis, Jr., and Virginia Torczon. Direct search methods on parallel machines. *SIAM Journal on Optimization*, 1 :448–474, 1991.
- [16] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4) :308–313, January 1965.
- [17] Tony Heap and David Hogg. Towards 3d hand tracking using a deformable model. In *In Face and Gesture Recognition*, pages 140–145, 1996.
- [18] A. Meijster, J.B.T.M. Roerdink, and W. H. Hesselink. A general algorithm for computing distance transforms in linear time. In *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 331–340. Kluwer, 2000.