A NEW DEPTH-BASED FUNCTION FOR 3D HAND MOTION TRACKING

Ouissem Ben-Henia and Saida Bouakaz
LIRIS CNRS UMR 5205, Université Claude Bernard Lyon 1
43 Boulevard du 11 novembre 1918, 69622 Villeurbanne, France

Keywords: Hand tracking, Dissimilarity function, Minimization method.

Abstract: Model-based methods to the tracking of an articulated hand in a video sequence generally use a cost function to compare the hand pose with a parametric three-dimensional (3D) hand model. This comparison allows adapting the hand model parameters and it is thus possible to reproduce the hand gestures. Many proposed cost functions exploit either silhouette or edge features. Unfortunately, these functions cannot 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 stemming from depth maps. 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. To reduce the computational burden, we propose to compute a volume of voxels from a hand point cloud, where each voxel is characterized by its distance to that cloud. When we place a model point cloud inside this volume of voxels, it becomes fast to compute its distance to the hand point cloud. Compared with other well-known functions such as the directed Hausdorff distance(Huttenlocher et al., 1993), our proposed function is more adapted to the hand tracking problem and it is faster than the Hausdorff function.

1 INTRODUCTION

Our research focuses on complex hand motion tracking, with the aim of developing vision-based approaches capable of reproducing the hand gestures. These last years, research works on hand motion tracking have been strengthened, especially due to the growing need of human-computer interactions for entertainment and video games. Many vision-based approaches have been proposed to solve the hand tracking problem. According to the considered application, the existing methods could be grouped into two categories: view-based and model-based.

View-based methods use a database of predefined hand poses, which are generally recovered through classification or regression techniques(Rosales et al., 2001), (Shimada et al., 2001). A set of hand features is labeled with a particular hand pose, and a classifier is trained from this data. Due to the high dimensionality of the space spanned by possible hand poses, it is difficult, or even impossible to perform dense sampling. Therefore, when we consider a limited set of predefined hand poses, it becomes possible to cope with real-time applications such as a human-computer interaction one (Ike et al., 2007).

Model-based methods use a parametric three-dimensional (3D) hand model and provide more precise and smooth hand motion tracking(de La Gorce et al., 2008), (Stenger et al., 2006), (Kerdvibulvech and Saito, 2009), (Henia et al., 2010). The 3D hand model is often represented as a hierarchical one with approximately 26 degrees of freedom (DOF) (Kato and Xu, 2006), (Wu et al., 2005). Its appearance part is provided by an underlying geometric structure. To achieve hand motion tracking, these methods estimate the hand model parameters which reproduce the hand motion appearing in video sequences. For this purpose, a cost function is defined to compare the hand poses with the model ones. The well-know cost functions exploit either silhouette or edge features extracted from images shot by affordable cameras (Stenger et al., 2001), (Kato and Xu, 2006). These functions are fast to be computed but cannot deal with the tracking of complex hand motion such as the closing of hand. To solve this problem, other alternatives have proposed to use a color glove (Wang and Popović, 2009) which eliminates ambiguities between the palm down pose and the palm

Ben-Henia O. and Bouakaz S..
A NEW DEPTH-BASED FUNCTION FOR 3D HAND MOTION TRACKING.
DOI: 10.5220/0003378106530658
In *Proceedings of the International Conference on Computer Vision Theory and Applications* (VISAPP-2011), pages 653-658
ISBN: 978-989-8425-47-8

up one. Another depth map based-function was proposed in (Bray et al., 2004a). To speed up the tracking, the hand is subsampled at 45 stochastically determined points (2 for each visible phalanx and 15 for the palm). This simplification cause a loss of information.

In this paper, we propose a new depth-based function which makes use of all the hand points appearing in depth map. Our proposed function evaluates a distance between two clouds of points, which represent respectively the hand and its underlying 3D model. 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 distance computation by means of well-known functions such as the Hausdorff(Huttenlocher et al., 1993) one is computationally expensive. To reduce the computational burden, we propose to compute a volume of voxels from a hand point cloud, where each voxel is characterized by its distance to that cloud. Once this volume of voxels is obtained, and when we place a model point cloud inside this volume, it becomes fast to assess its distance to the hand point cloud. The remainder of this paper is organized as follows. The next section presents the used 3D hand model. Our proposed function is detailed in section 3. Section 4 describes the hand motion tracking algorithm. Before concluding, experimental results from synthetic data are presented in section 5.

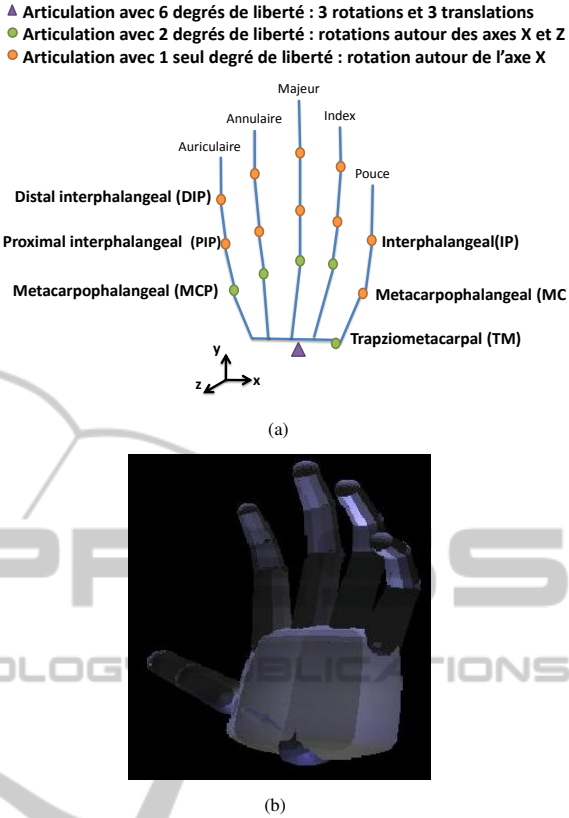


Figure 1: (a) Squeletal representation of the 3D model (b) 3D hand model appearance.

2 THE HAND MODEL

The human hand is a complex and highly articulated structure. Several models have been proposed in the literature. In (Heap and Hogg, 1996) a 3D deformable point distribution model was implemented. This model can not accurately reproduce all realistic hand motion. Indeed, since this model is not based on a rigid skeleton, fingers can be warped and reduced to ensure tracking of the hand gestures.

On the other hand, skeleton animation based model was used in (Bray et al., 2004b), (Ouhaddi and Horain, 1999). This kind of models is usually defined as hierarchical transformations representing the DOF of the hand: position and orientation of the palm, joint angles of the hand(Figure:1(a)). The variation in the values of DOF animates the 3D hand model. Using this kind of models, we can estimate not only the position and orientation of the hand, but also the joint angles of fingers.

In our proposed work, we use a parametric hand model which is conforming to the H-Anim standard[1]. The H-Anim model is often used in the 3D animation field. We can highlight its particularity to separate the kinematic part (motion) from the appearance one. This model consists of a hierarchy of 3D transformations (rotation, translation) allowing easy control of its animation by modifying only the involved transformations. Regarding the appearance, objects called segments are placed in this hierarchical representation to provide the shape of this model (Figure 1(b)). To change the appearance of this model, we modify the objects representing the model apprearence. In our proposed work, these object segments are modeled by quadric surfaces as shown in Figure 1(b). In our hand model, each finger is modeled as a four DOF kinematic chain attached to the palm. Together with the position and orientation of the palm, there are 26 DOF to be estimated.

For the model appearance, we use a set of quadrics approximately representing anatomy of a real human

[1]Humanoid-Animation(H-ANIM) is an approved ISO standard for humanoid modeling and animation. website :www.h-anim.org

hand (Figure 1(b)). The palm is modeled using a truncated ellipsoid, its top and bottom closed half-ellipsoids. Each finger is composed by three truncated cones, i.e. one for each phalanx. Hemisphere was used to close each truncated cone. The major advantage of this model shape representation is its simplicity to be adapted to any hand to track compared with models based on 3D scans.

3 DISSIMILARITY FUNCTION

We propose to assess a distance between two clouds of points, which represent respectively the hand and its underlying 3D model. Two well-known functions are often used to compute distance from a set of points A to another one B. The first one is given by the following formula:

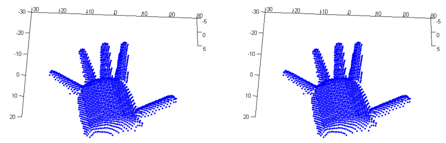
$$d_1(A,B) = \frac{1}{|A|} \sum_{a_i \in A} \min_{b_j \in B} d(a_i, b_j) \qquad (1)$$

where $|A|$ is the cardinal number of the set of points A, and $d(a_i, b_j)$ the Euclidean distance between the two points a_i and b_j. The second well-known function is the directed Hausdorff distance. We denote this function d_2.and we present its by means of the following equation :

$$d_2(A,B) = \max_{a_i \in A}\{\min_{b_j \in B} d(a_i, b_j)\} \qquad (2)$$

where the Euclidean distance $d(a_i, b_j)$ is identical to the one described above in Eq.1 The distance computation by means of this class of functions is performed in a non-linear way. Indeed, the point cloud B is sweeped as many times as points in A. In our case study, these functions are computationally expensive due to the fact that for each frame several distances are to be computed. These distances are defined from the same cloud of points of the hand to several point clouds obtained by the hand model. In this paper we propose a well suited function to our case study. This function is based on 3D distance transform and performed in two stages. The first one computes a volume of voxels from the cloud of points A, where each voxel encompasses the distances of the voxel to A. The cubic voxel is used in the second stage to evaluate the distance from the point cloud A to another one B.

The algorithm developed to compute the voxel volume is inspired from the method proposed by A. Meijster et all in(Meijster et al., 2000), where a distance map is computed from a 2-D (bidimensional) image. This algorithm could be extended to an n-D(n-dimensional) space. We implement an extension of this algorithm for the 3-D (three-dimensional) space to compute a cubic volume *CV* (Figure 2(b)) from the point cloud A (Figure 2(a)). In Figure 2(b) color corresponds to distance value. More detailed could be found in(Meijster et al., 2000).



(a) (b)

Figure 2: (a) Point cloud A(b) A voxel volume obtained from the point cloud A.

Once the 3-D volume is obtained, we can compute the distance of any point cloud B to the cloud A using the following formula:

$$F(A,B) = \frac{1}{|B|} \sum_{b_i \in B} CV[b_{ix}][b_{iy}][b_{iz}] \qquad (3)$$

where b_{ix} b_{iy} b_{iz} are the coordinates of the point b_i.

4 THE TRACKING ALGORITHM

The tracking of the hand gestures in a video sequence is performed by seeking the hand model parameters which reproduce the hand motion as summarized in Figure 3. We achieve this step by minimizing our dissimilarity function for each frame of the video sequence. This minimization provides the hand model parameters which align the model pose with the hand one. We assume that the hand model parameters are close to the solution associated with the first frame of the video sequence. For the remainder of the video sequence, the minimization process exploits the hand model parameters obtained at the previous frame. We use the iterative algorithm proposed by Torczon(Dennis et al., 1991) to explore different directions for each iteration and keep the one minimizing the dissimilarity function. The Torczon's algorithm is an amelioration of the Nelder and Mead's one (Nelder and Mead, 1965).

The choice of the Torczon's algorithm is supported by two main facts. Firstly, the use of this algorithm does not require the knowledge of the derivative of the function to be minimized. The second fact relates to the processing of the Torczon's algorithm itself. Indeed, the Torczon's algorithm explores different directions for each iteration. These various explorations can be achieved in parallel to reduce the computational burden.

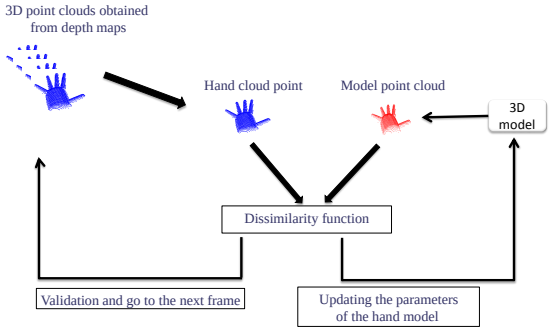


Figure 3: Tracking process.

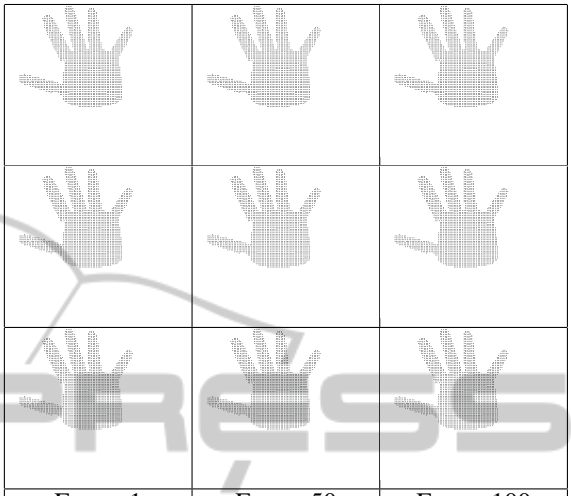
Due to the high number of the model parameters to be estimated, we use a simplification proposed in (Henia et al., 2010) to minimize the dissimilarity function in two steps. The first one estimates the position and orientation of the hand. The parameters representing the joint angles of fingers are fixed, whereas those representing the position and orientation of the palm are processed by the minimization algorithm. The processing is reversed at the second step, i.e. the orientation and position parameters are fixed to those obtained in the first step, and the joint angles of the hand are estimated. Besides simplifying the minimization problem, this approach can be supported by the slow variation of the hand pose in two successive frames.

5 EXPERIMENTAL RESULTS

The performances achieved by our proposed work are evaluated for tracking hand motion appearing in synthetic images. A video sequence of one hundred 320x240 synthetic images of the hand model is acquired (Table 1). To obtain this sequence of images, we vary the parameters of each finger except the thumb and the index. Through this test benchmark we can highlight the importance of the depth information to cope with complex hand motion. Our proposed function compares point clouds generated from depth maps obtained by means the OpenGL library.

We compare the results obtained by our proposed function with those achieved by means of the silhouette-based function proposed in (Henia et al., 2010). The silhouette-based function is not adapted to the tracking of abduction finger motion. Indeed, the finger motion tracking is lost at the frame 50 of the video sequence (Table 1). To estimate the error of the tracking algorithm, we compute a difference between the tracking results and the ground truth. The tracking error is then plotted as a curve in Figure 4, in which we only consider the PIP(Figure 1(a)) joint angles of

Table 1: Tracking results using synthetic data : the first line represents 3d point clouds corresponding to the ground truth, the second line shows the tracking results using a silhouette-based function, the third line represents the tracking results using our proposed depth-based function.



Frame 1	Frame 50	Frame 100

the middle finger. With our proposed depth-based function, the error is very small when it is compared with another silhouette-based function (Henia et al., 2010). Indeed, the average tracking error drops from 0.5 radian with the silhouette-based function to 0.1 radian with our proposed depth-based function (Figure 4). Our observation concerning the tracking error of the PIP joint angle of the middle finger could be still valid for other parameters.

Using the same test benchmark (Table 1), the results obtained by our proposed function are also compared with those achieved by means of the Hausdorff function. The tracking results are very close in terms of accuracy but are very different regarding computing time. The running time is about 3 seconds per frame using our proposed function, whereas 300 seconds per frame are required by the Hausdorff function. We can highlight that the average cardinal of the point clouds used in our test benchmark is approximately 3000 points. We can see that our proposed function is well-adapted to the hand tracking problem because the computing time is acceptable even if the point cloud is significant. The processing is performed using a PC Intel-Centrino 2 GHZ processor and Nvidea graphic card (GeoForce 8600MGT).

6 CONCLUSIONS AND FUTURE WORK

This paper presents a new depth-based function that

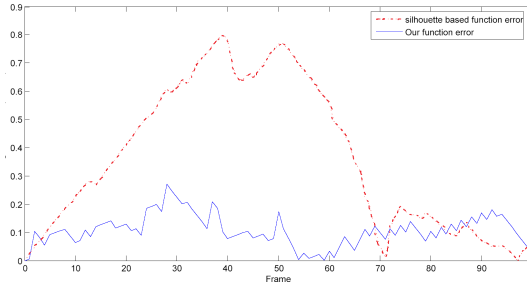


Figure 4: Tracking error of the PIP middle finger.

is well-adapted to the 3D hand tracking. A 3D parametric hand model is used to achieve the tracking by comparing its poses with the hand ones using our proposed function. The depth-based function compares 3D point clouds stemming from depth maps. 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. Classical functions comparing 3D point clouds such as the Hausdorff one are not adapted for the hand tracking problem because of the expensive time needed to achieve both the comparison and tracking. To reduce the computational burden, we propose to compute a volume of voxels from the hand point cloud, where each voxel is characterized by its distance to that cloud. By placing any model point cloud in the computed volume, it becomes fast to compute its distance to the hand point cloud. We experiment our proposed function using synthetic data obtained from depth maps generated by means of the OpenGL library. The preliminary results obtained so far are very encouraging because we are able to track complex hand motion such as the closing of hand. Besides tracking complex hand motion, our proposed function is faster than other well-known functions such as the Hausdorff one.

We plan to extend our experimental study using real data. For this propose, different methods could be used to collect 3D hand point cloud. Stereo vision could be a solution to the problem of acquiring 3D cloud point of the hand but it requires the use two video cameras. Another alternative consists to use a new generation of video cameras, called time of flight cameras, and provides a 3D cloud point of the observed scene in real-time. However, this new technology is deemed to be not very precise. Structured light sensor could also be used to obtain depth maps of the hand as it is done in (Bray et al., 2004a). This method seems to provide accurate results but it is slower than a time of flight camera. A comparative study between these different methods must be performed to select the one yielding the best results in terms of accuracy and computing time.

REFERENCES

Bray, M., Koller-meier, E., Mller, P., Gool, L. V., and Schraudolph, N. N. (2004a). 3d hand tracking by rapid stochastic gradient descent using a skinning model. In *In 1st European Conference on Visual Media Production (CVMP*, pages 59–68.

Bray, M., Koller-Meier, E., Schraudolph, N. N., and Gool, L. V. (2004b). Stochastic meta-descent for tracking articulated structures. In *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 1*, page 7, Washington, DC, USA. IEEE Computer Society.

de La Gorce, M., Paragios, N., and Fleet, D. J. (2008). Model-based hand tracking with texture, shading and self-occlusions. In *CVPR*.

Dennis, J. E., Jr., and Torczon, V. (1991). Direct search methods on parallel machines. *SIAM Journal on Optimization*, 1:448–474.

Heap, T. and Hogg, D. (1996). Towards 3d hand tracking using a deformable model. In *In Face and Gesture Recognition*, pages 140–145.

Henia, O. B., Hariti, M., and Bouakaz, S. (2010). A two-step minimization algorithm for model-based hand tracking. In *WSCG*.

Huttenlocher, D. P., Klanderman, G. A., Kl, G. A., and Rucklidge, W. J. (1993). Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:850–863.

Ike, T., Kishikawa, N., and Stenger, B. (2007). A real-time hand gesture interface implemented on a multi-core processor. In *MVA*, pages 9–12.

Kato, M. and Xu, G. (2006). 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.

Kerdvibulvech, C. and Saito, H. (2009). Model-based hand tracking by chamfer distance and adaptive color learning using particle filter. *J. Image Video Process.*, 2009:2–2.

Meijster, A., Roerdink, J., and Hesselink, W. H. (2000). 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.

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

Ouhaddi, H. and Horain, P. (1999). 3d hand gesture tracking by model registration. In *Proc.IWSNHC3DI99*, pages 70–73.

Rosales, R., Athitsos, V., Sigal, L., and Sclaroff, S. (2001). 3d hand pose reconstruction using specialized mappings. In *ICCV*, pages 378–385.

Shimada, N., Kimura, K., and Shirai, Y. (2001). 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. IEEE Computer Society.

Stenger, B., Mendonca, P. R. S., and Cipolla, R. (2001). 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.

Stenger, B., Thayananthan, A., Torr, P. H. S., and Cipolla, R. (2006). Model-based hand tracking using a hierarchical bayesian filter. *IEEE Trans. Pattern Analysis and Machine Intelligence(PAMI)*, 28(9):1372–1384.

Wang, R. Y. and Popović, J. (2009). Real-time hand-tracking with a color glove. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, pages 1–8, New York, NY, USA. ACM.

Wu, Y., Lin, J., and Huang, T. S. (2005). Analyzing and capturing articulated hand motion in image sequences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(12):1910–1922.