POLITECNICO DI TORINO
Repository ISTITUZIONALE

A fast and robust hand-driven 3D mouse

Original
A fast and robust hand-driven 3D mouse / Bottino, A.G., DE SIMONE, M.. - (2009), pp. 567-574. (VISAPP 2009 Lisboa, Portugal 5-8 February 2009).

Availability:
This version is available at: 11583/1853291 since:

Publisher:

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

10 September 2026

Author's version

Published in: VISAPP 2009 - Proceedings of the 4th International Conference on Computer Vision Theory and Applications, Volume 1, 2009, Pages 567-574

A FAST AND ROBUST HAND-DRIVEN 3D MOUSE

Andrea Bottino, Matteo De Simone
Dipartimento di Automatica e Informatica, Politecnico di Torino,
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
andrea.bottino@polito.it, matteo.ds@gmail.com

Keywords: Computer vision, image processing, 3D hand tracking, non intrusive motion capture, real time tracking

Abstract: The development of new interaction paradigms requires a natural interaction. This means that people should be able to interact with technology with the same models used to interact with everyday real life, that is through gestures, expressions, voice. Following this idea, in this paper we propose a non intrusive vision based tracking system able to capture hand motion and simple hand gestures. The proposed device allows to use the hand as a "natural" 3D mouse, where the forefinger tip or the palm centre are used to identify a 3D marker and the hand gesture can be used to simulate the mouse buttons. The approach is based on a monoscopic tracking algorithm which is computationally fast and robust against noise and cluttered backgrounds. Two image streams are processed in parallel exploiting multi-core architectures, and their results are combined to obtain a constrained stereoscopic problem. The system has been implemented and thoroughly tested in an experimental environment where the 3D hand mouse has been used to interact with objects in a virtual reality application. We also provide results about the performances of the tracker, which demonstrate precision and robustness of the proposed system.

1. INTRODUCTION

In the recent years there has been a growing interest in the scientific and industrial communities to develop innovative devices allowing a natural interaction with machines, and recent products, like the Nintendo Wii console and the Microsoft Surface, try to introduce novel habits for computer users. From this point of view, one of the most natural ways to interact with objects is using our bare hands. In real world, we use our hands to touch, grasp and move the objects and we typically use our fingers to point at something. Acting in the same way on the objects, allows people to transfer already acquired abilities to the interaction with the computer and to expand possibilities and complexities in human-to-computer communication.
Several commercial gesture based devices exist at present. They usually require extraneous wearable hardware, such as markers or sensing gloves. These solutions provide high quality results but they are expensive, intrusive and they could disturb the movements of the user.
On the contrary, vision based techniques could offer a simple, expressive, and meaningful manner to interact with the computer. These techniques are cost-effective and non invasive, and they have been used in many contexts.
In this paper, we propose a simple, fast and robust hand tracking system that can be used to develop a "natural" 3D mouse. The output of the tracker is the position of a 3D marker and an indication of the actual posture of the hand, which can be used to mimic the click of the mouse buttons. Our system uses two video cameras and the proposed approach separates the elaboration of the two image streams, refining the results in a merging stage at

the end of the process. In more details, for each image stream, we evaluate the perceived hand posture and the 2D position, on the image plane, of the reference point. The results are then combined in order to evaluate the final 3D marker position and hand posture.
Experimental results demonstrate that the proposed system is fast, robust and computationally manageable on medium level computers. The paper is organized as follows. In Section 2, we describe our approach. Section 3 evaluates the experimental results. Concluding remarks are reported in Section 4.

1.1 Related works

The general approach to vision based hand tracking usually requires three main processes: hand segmentation, hand tracking and gesture estimation. Colour is the most common image clue for feature extraction (Bradski (1998), Chen, Fu and Huang (2003), Pantrigo, Montemayor and Sanchez (2005)). Other visual cues, like motion, edges and shading have been proposed in order to reduce influences of varying illumination and cluttered background (Cui and Weng (2000), Liu and Jia (2004)). Recent results in integrating different visual information offer more robust solutions (Akyol and Alvarado (2001), Shan Lu et al (2003)).
Several approaches have been applied to the tracking problem: a review of the most popular algorithms is presented in Mahmoudi and Parviz (2006). The CAMShift algorithm (Bradski, 1998) is a robust nonparametric technique derived form Mean Shift (Cheng, 1995). Improvements of CAMShift algorithm can be found in Liu et al. (2003), Zhai et al. (2005). The Condensation algorithm (Isard and Blake, 1998) is a powerful stochastic approach based on Monte Carlo method that has been applied to several challenging environments, such as wearable computers (Liu and Jia (2004)) and real-time humanoid interaction (Gumpp et al, 2006). Improvements of Condensation are Icondesation (Isard and Blake, 1998), particle filtering (Weiser and Brown (1995), Shan et al (2004)), smart particle filtering (Bray, Koller-Meier and Van Gool, 2004) and local search particle filter (Pantrigo, Montemayor and Sanchez, 2005). Articulated and deformable 3D hand model driven techniques have also been proposed by Stenger, Mendonca and Cipolla (2001), Heap and Hogg (1996).
As for the gesture estimation problem, a good survey on the subject is Erol et al. (2007). We can have partial pose approaches (Oka, Sato and Koike (2002), Letessier and Bèrard (2004)), which are usually based on rough models of the hand and cannot reconstruct all the degrees of freedom (DOF) of the hand, and full DOF approaches, usually exploiting complex 3D models (Drummond and Cipolla (2002)), and performing single frame pose estimation offering the advantage of self-initializing and re-initializing algorithms, capable of handling fast hand motion where time coherence is an useless clue (Stenger et al, (2004), Tomasi, Petrov and Sastry, (2003)).

2. THE MULTIPLE CAMERA APPROACH TO HAND TRACKING

The goal of our work is to develop a simple and effective input device which allows a “natural” way of interacting with the computer using the bare hands. In our scenario, a single user sitting at a desktop interacts with a VR application picking 3D points, selecting and moving objects in the environment or changing the view on the simulated world. To perform these tasks, the 3D counterpart of a desktop mouse, that is a device capable of moving a 3D marker in the environment and to issue commands by pushing some buttons, could be sufficient. From the point of view of the user, which is immersed in a virtual environment where the objects are floating in front of his eyes, the most natural way to select an object would be by touching it, to change its position by moving it with the hand, and picking a location by pointing it. To select a different view of the scene, a commonly used approach is the virtual trackball paradigm: the objects are enclosed in a glass ball which can be rotated around its centre to change the world view. Again, a natural way to perform this operation would be to touch the glass ball and rotate it with the hand. Therefore, the position of the forefinger tip can be used as a “natural” 3D marker that can be controlled moving the hand (Figure 1a-b). A simple way to simulate the click of the mouse could be to associate the event to a specific, and comfortable, hand posture. For instance, when the forefinger is extended, such a posture could be the one where also the thumb is stretched out (Figure 1c-d). Thumb-index enslaving is not a problem for ergonomy, since Olafsdottir (2005) shows that the indices of digit interaction does not depend whether the thumb is one of the involved digits.

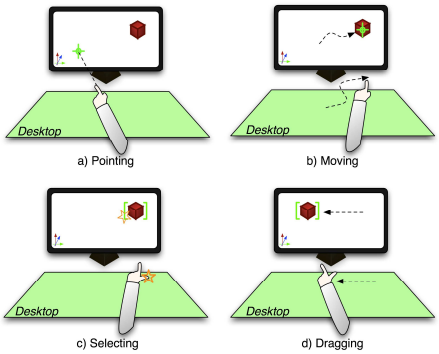

Figure 1: interaction with the virtual environment

The click of a second button, e.g. the right button, could then be mimicked with another posture. We experienced that extending another finger is not comfortable for the user, while it is much easier to close the hand. Using this posture, the forefinger tip cannot be used anymore as 3D marker, and another reference point, for instance the centre of the palm, can be chosen.
The proposed hand tracking system uses two cameras to reconstruct the (x,y,z) position of the marker and of the hand posture. The two cameras are located on the sides of the desktop where the user sits and are positioned and oriented in order to observe a common area, the active area, where the user can move its hand.
Our approach works in two phases. First, the information about 2D marker position and perceived hand posture are extracted from each image stream separately. Then, these data are combined in order to obtain precise 3D information. The rationale of this approach is that the 2D information can be obtained in a fast and reliable way, and that processing separately the two image streams allows the two threads to be executed in parallel on multi-cpu or multi-core based architecture, guaranteeing a substantial improvement of the execution times.
The result is a simple and computationally fast system, which is very robust against noise and cluttered backgrounds. The outline of the proposed hand tracking system is the following.
After initializing some of the parameters used by the system, the two image streams are processed separately. This step involves three processes:

1. detection/tracking: identifies in the incoming image the user hands and tracks it along the video sequence
2. segmentation: extracts the silhouettes of the hand from the incoming image
3. recognition: identifies the posture of the hand and the 2D marker position

Finally, stereoscopy is used to combine 2D information and to obtain the required 3D marker posture and hand gesture. In the following, the single components of these processes will be described in details.

2.1 Initialization

Initialization is an offline process that involves several system's parameters.
First, the camera needs to be calibrated. This process establishes a relation between the image planes of the cameras and a fixed reference system in the 3D world. In this work we used the approach described in the Open Computer Vision Library (n.d.).
Second, since the tracking algorithm uses a probability distribution image of the chromatic components of the objects to be tracked, a reference colour model needs to be initialized. In this case, we are interested in detecting skin-like pixels. It has been demonstrated (Bradski, 1998) that different skin colour models are not needed for different races of people. This fact allows building a priori a skin model that can be used to identify the hand. This is modelled with a simple chromaticity histogram, created interactively by selecting a part of the image corresponding to the hand. The RGB values of the pixels of the selected region are converted into the Hue Saturation Value (HSV) colour system, allowing to create a simple colour model taking 1D histograms from the hue channel.

2.2 Single-stream processing

Single stream processing is based on the approach presented by Bottino and Laurentini (2007). It combines several computer vision algorithms, in order to exploit their strengths and to minimize their weakness.

2.2.1 Detection/tracking

The detection/tracking module is a state machine, whose state diagram is shown in Figure 2. In the detection state, the input image is processed until a hand enters the image. Then, in the tracking state, the hand is tracked until it exits the image.

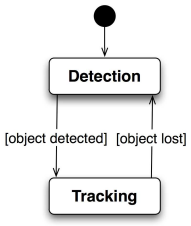

Figure 2: detection/tracking module

This module uses two different algorithms: Mean shift (Cheng, 1995) for object detection and CAMShift (Bradski, 1998) for object tracking

The input of both algorithms is a probability image, where pixels more likely to belong to the searched object have a higher value. This probability image is obtained by back projecting on the image the chromaticity histogram evaluated during the initialization step. The probability image is then filtered, as suggested by Bradski (1998), in order to ignore pixels whose hue value is not reliable. An example of an image frame and its corresponding probability image is shown in Figure 3a and Figure 3b.

Mean shift uses a search window, whose size is fixed, which is iteratively moved along the image "climbing" the probability distribution until the optimum is reached. At each step, the centre of mass of the search windows is evaluated and the search window is centred on it. In our implementation, in order to avoid systematic errors, the initial position of the search window is randomly set in several positions of the image. The final location giving the best score is chosen (Figure 3c and Figure 3d). To find when an object is detected, the percent of object-like pixels in the search window is compared with a pre-defined threshold.

CAMShift is a generalization of the Mean shift. While Mean shift is designed for static distributions, CAMShift is designed for dynamically changing distributions, such as those occurring in video sequences where the tracked object moves, so that size and location of the probability distribution change in time. Hence, at every iteration, also the size of the search windows is adapted to the incoming distribution. Again, to find when an object is lost, the percent of object-like pixels in the search window is compared with a pre-defined threshold.

The output of this module is a flag indicating if the object has been detected and the region R where it is located.

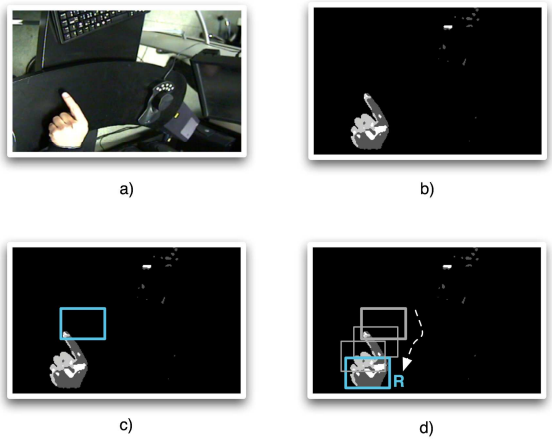

Figure 3: a) An incoming image and b) the corresponding histogram back projection. c) The initial detection region. d) The Mean Shift moves the R region to the optimum.

2.2.2 Segmentation

The segmentation module, given position and dimension of R, extracts the silhouette of the hand from the probability image. First, the probability image is thresholded in order to obtain a binary image. Then, morphological operators are applied to remove spurious pixels and holes are removed with a flood fill algorithm. Finally, the bounding box of the main connected component contained in the search window is evaluated. Any further processing on the images will take place only on this Region of Interest (ROI), reducing the computational burden. Other disturbing connected components, not belonging to the hand, are discarded (Figure 4).

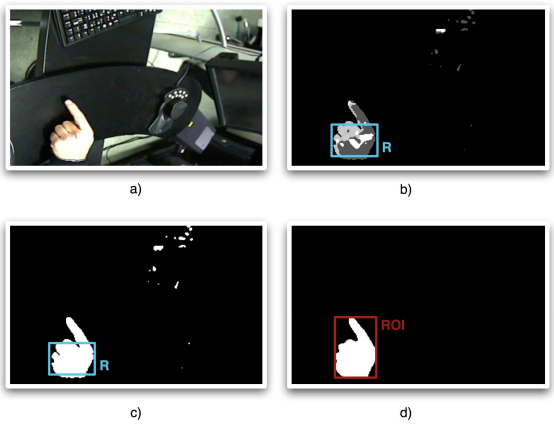

Figure 4: segmentation process. a) source image, b) probability image and R region, c) thresholded image, d) the main connected component in R and the correspondng ROI.

2.2.3 Recognition

The input of this process is the silhouette of the moving object and the R and ROI regions. The output is the 2D position of the marker and the posture of the hand.
A simple 3D model of the hand is used to reconstruct the desired information. It is composed by one to three ellipsoids, depending on the gesture to represent. Each ellipsoid is represented in matrix form as x'Qx, where x' = [*x y z 1*] and Q is a coefficient matrix. Using this representation, every transformation (translation, rotation, scaling) can be applied to the model with a simple matrix multiplication. The model has 7 degrees of freedom, 3 for the position, 3 for the orientation and 1 for the posture, which determines also the number of ellipsoids composing the model. The posture can assume three discrete values: 0, hand closed 1, hand closed with the

forefinger extended, and 2, hand closed with thumb and forefinger extended. The three postures, the shapes of the corresponding models and the assigned values are shown in Figure 5. The projections of the ellipsoids on a plane are quadrics and can be obtained, again, with a simple multiplication between matrices. Then, knowing the projection matrix obtained from calibration is sufficient to project the model on the image plane.

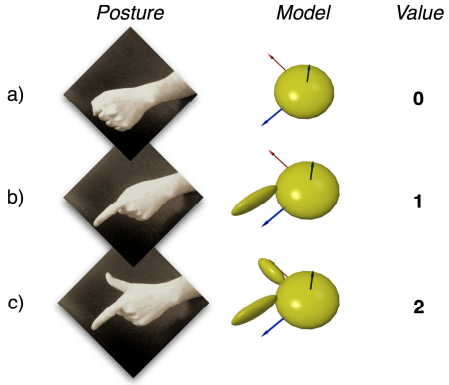

Figure 5: the three hand postures identifiable, the corresponding model shape and the assigned value.

The state of the model is reconstructed using the ICondensation algorithm (Liu and Jia, 2004), which is a combination of the statistical technique of importance sampling with the Condensation algorithm (Isard and Blake, 1998). Basically, it tries to approximate the unknown statistic distribution of the state of a process with a discrete set of samples and associated weights. This set evolves iteratively from an initial set of random samples, and at each step t their weights are updated. From these weights, it is possible to predict the probability distribution over the search space at time t+1. Thus, samples with higher probability are more likely to be propagated over time. The process is iterated for a predefined number of steps, and the final state is given by the weighted sum of the final samples. In our case, each sample represents a model state and the weighting function is defined from the projection of the model on the image plane. Given I_s, the result of the exor of model projection and silhouette of the hand, the weight of the sample s is given by:

$$w_s = \frac{1}{1+\sum_{(x,y)\in I_s} I_s(x,y)} \qquad (1)$$

The initial set is created from random samples in the neighborhood of an initial guess of the hand state. This is obtained from an analysis of the incoming silhouette, which works as follows. From the segmentation process we have the R and ROI regions (Figure 6). The palm of the hand as a high probability to fall into R, while finger pixels are mainly in the area (ROI-R). Therefore, R can be used to extract 2D position and orientation of the palm.

The first order moments of R give a reasonable indication of the centre of the palm, while dimension and orientation of the ellipse corresponding to the palm can be deduced from the covariance ellipse built on the second order moments. From these parameters, we can obtain, given the dimension of the user hand, a rough approximation of the distance from the camera and of the orientation of the ellipsoid corresponding to the 3D palm. An initial indication of the hand gesture can be obtained analyzing the moments of the ROI region. In particular, the third order moments give us an indication of the asymmetry of the image along its principal axes. A significant asymmetry along the major axis is a strong indication of the forefinger presence, while a significant asymmetry along the minor axis is an indication of the thumb presence.

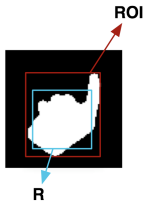

Figure 6: ROI, external rectangle, and R, inner rectangle

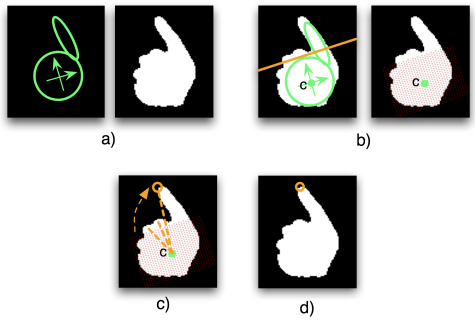

Figure 7: Marker identification. a) Input data: model (left) and silhouette (right). b) The model is used to mask the palm. c) Curvature evaluation of boundary points. d) The marker is located on the point with maximum curvature.

The output of Icondensation is then used to identify precisely the 2D position of the marker. If the posture is 0 (closed hand), the marker is given by the centre of the projection of the 3D palm. Otherwise, we identify the fingertip projecting the palm on the image and using it to mask the region corresponding to the forefinger. In case the posture is 2, the connected component of interest is the one overlapping the projection of the 3D ellipsoid corresponding to the forefinger. Identification of the marker position is based on the analysis of the local curvature of the boundary of the obtained region. The farthermost point at maximal curvature away from the centre of the projection of the palm, is taken as the 2D position of the fingertip. An example can be seen in Figure 7.

2.2.4 Merging

The merging process involves two steps:

- a reconstruction step, where the 3D position of the marker and the hand posture are reconstructed on the base of the information evaluated from the left and right images
- a data filtering step, where a Kalman filter is used to smooth the noise in the extracted data

Regarding the first step, the hand posture is evaluated by means of the following voting scheme:

Table 1: the voting scheme used to evaluate the hand posture.

		Left Camera Pose		
		0	***1***	***2***
Right Camera Pose	***0***	0	1	2
	1	1	1	2
	2	2	2	2

As a matter of facts, one or more fingers can be hidden in one image, but visible in the other. If the posture number indicates the number of extended fingers, our voting scheme will choose the maximal value identified for left and right posture.

When the posture is 0 or when the forefinger is visible in both images, the 3D marker position is evaluated intersecting the lines back projection in 3D the marker positions on the image planes from the corresponding optical centres. The line equation is evaluated by pseudo inverting the projection matrix. Due to the noise introduced by the previous reconstruction processes, the lines unlikely intersect each other. So we choose as intersection the point at minimal distance from both lines.

When a reference point, such as the fingertip in our example, is visible in only one image, the marker position can be reconstructed in the following way. First, we assume that the forefinger tip lies on the principal plane of symmetry of the hand. Then we extract the orientation of this plane, P, from the model corresponding to the image where the forefinger is visible. Finally, we take the plane P' parallel to P and passing through the 3D position of the palm centre and we intersect it with the line back projecting the visible forefinger tip.

3. SYSTEM EVALUATION

The proposed hand tracking device has been implemented and tested on a PC with an Intel DualCore E6600 CPU, 1GB of RAM. The frame grabber used for image capturing is a cost-effective PCI capture board and guarantees an overall 240 color frames per second. The system includes two color cameras, with maximal frame rate of 50 fps, and 752x582 color images. According to the literature on the subject and referring to the available commercial products, the device can be evaluated according to a set of desirable characteristics.

3.1 Robustness

For robustness we mean the quality of being able to cope well with variations in the operating environment with minimal loss of functionality.

Initialization and re-initialization
Since the hand is continuously entering and exiting the active area, the system must guarantee a robust identification of the hand presence. In a single image stream, when the hand is tracked, the CAMShift algorithm provides useful information on the object identified in order to understand when it exits the image. At the same time, Mean shift can recover very efficiently the object as soon as it enters again the image. Distributing casually the search window over the image allows easily to “hook” the object and then to identify its position. Therefore, we can state that the system provides robust initialization and re-initialization of the tracking components at thread level, which is reflected into the robustness of the two integrated streams.

Cluttered backgrounds
The system is not sensible to non uniform backgrounds or moving objects, unless their chromatic distribution is not similar to the one of the tracked object. Examples can be seen in Figure 3 and Figure 4, where a complex background is present. Some small groups of pixels not belonging to the hand can be present in the probability image, but they are discarded during segmentation. For skin-like objects entering the image, such as leather tissues, we stress that the CAMShift algorithm is very robust against distracters. Once CAMShift is locked onto the mode of a colour distribution, it will tend to ignore other nearby but non-connected distributions.
It is true, however, that some problems can be caused when the disturbing object and the hand form a connected component or when a disturbing object, whose area is bigger then the identification threshold, is detected in the image when no hand is present. This produces false hand identification.

Independence from illumination
The system is guaranteed to work for a wide range of variations of illumination intensity since the segmentation process is theoretically independent from the illumination conditions. However, if the global illumination falls below a certain threshold, the segmentation algorithm does not give good results anymore. The same problem happens when the global level of illumination is too high, for instance for direct sunlight hitting the working area, since the camera saturates.

3.2 Computational manageability

The machine used during the test can be considered as a medium cost processing unit (the complete system, including capture board and camera has a cost lower than 1.500€). Processing the image streams at a frame rate of 25 frames/sec, the mean latency is 20ms, and each single CPU is used at 30% of its capacity. The maximal mean update rate of 50 updates/sec. A faster computational unit guarantees to run the system reliably at 50 Hertz, which is adequate to the frequencies of normal gestures in human-computer interaction.

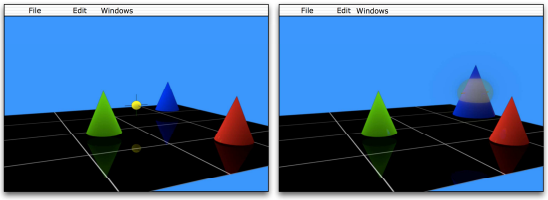

Figure 8: a) The test application. b) selecting and moving 3D objects

We developed, for test purpose, a simple VR application shown in Figure 8. The user can move the pointer (represented by a sphere), select and drag objects around the virtual world and change the user's view (see Figure 1 for a schematic representation). When the VR application is running another 25% of a single core is used. This demonstrates that there are resources available for other tasks, and that the system can be effectively used as an input device for other applications.

3.3 Performances of the HT system

Ground truth data was used to derive a quantitative measure of tracker's accuracy. Taken a grid of points whose 3D position is known, the accuracy can be evaluated through the difference between the 3D positions reconstructed from the 2D projections of the reference points and the ground truth data. Since the accuracy is not constant, the reference points must be located throughout the active area, in order to evaluate a mean accuracy value. As a result, we obtain an accuracy of 1.40 mm RMS.
The resolution is the minimal difference between two 3D positions detectable as different from the tracker, that is when their projections on the image planes are different, the minimal distance being one pixel. Therefore the resolution is given by the ray of the sphere enclosing all the 3D points obtained by back projecting 2D image points at one pixel distance point with the reference projection. Also the resolution is not uniform in the active area and the punctual resolution is computed and averaged over the set of reference points. The obtained mean resolution is 0.79 mm RMS.
Jitter and drift can be computed by placing the forefinger in a fixed position and computing the standard deviation of the reference position. The resulting jitter is 0.46 mm RMS, and the drift is null. The last result comes from the fact that none of the algorithms used introduce any drift in its output. It should be also outlined the fact that the jitter is lower then the tracker resolution, producing a stable output.
The characteristics of the tracker are summarized in Table 2. These results show that the tracker provides a sufficient precision for many applications.

Table 2: Summary of tracker characteristics

Tracker data	spatial position (3) hand posture (1)	**Jitter**	0.4 mm RMS
		Drift	none
Accuracy	1.4 mm RMS	**Latency**	20 ms
Resolution	0.8 mm RMS	**Update Rate**	>25 Hz

4. CONCLUSIONS

In this paper we have proposed a novel hand tracking system that can be used as object manipulation interface to use the bare hand for navigation, selection and manipulation tasks in virtual environments. The system is non intrusive, and reconstructs the position of a 3D marker and simple hand postures using two separate image streams. Each stream is processed separately with a monoscopic algorithm, which is very robust against noise and cluttered backgrounds. This allows reducing computing times parallelizing the processing of the two streams on multi-core or multi-cpu machines. Information from the two image planes are then combined in order to reconstruct the required 3D data.
The system has been implemented, and the tests demonstrates that it can run in real time (up to 50 samples per second) on today's desktop computer using off-the-shelf hardware components. Moreover, the device uses only

the 30% of the computing resources, allowing the execution on the same machine of other applications. As for precision, the results are satisfactory, showing an accuracy of 1.4mm, a resolution of 0.8mm and a jitter of 0.4 mm.

At present, the system requires that only one hand is present in the active area. As future work, we are planning to expand the system in order to use both hands for interaction.

5. REFERENCES

Akyol S., Alvarado P., 2001. 'Finding Relevant Image Content for mobile Sign Language Recognition', *Proc. Signal Processing, Pattern Recognition and Application*, pp. 48-52

Bottino A., Laurentini A., 2007. 'How to Make a Simple and Robust 3D Hand Tracking Device Using a Single Camera', *CSCC 2007*, Agios Nikolaos, Greece.

Bray M., Koller-Meier E., Van Gool, L., 2004, 'Smart particle filtering for 3D hand tracking', *Proc. IEEE Intern. Confer. on Automatic Face and Gesture Recognition*, pp. 675 – 680

Bradski G. R., 1998. 'Computer Vision Face Tracking For Use in a Perceptual User Interface', *Intel Technology Journal* (2), pp. 215.

Chen F.-S., Fu C.-M., Huang C.-L., 2003. 'Hand Gesture Recognition Using a Real-Time Tracking Method and Hidden Markov Models', *Image and Vision Computing* vol. 21, August, pp. 745-758

Cheng Y., 1995. 'Mean shift, mode seeking, and clustering', *IEEE Trans. PAMI.*, vol. 17, pp. 790-799

Cui Y., Weng J., 2000. 'Appearance-Based Hand Sign Recognition from Intensity Image Sequences'. *Computer Vision Image Understanding*, vol. 78, February, pp. 157-176

Drummond T., Cipolla R., 2002. 'Real-time visual tracking of complex structures'. *IEEE Trans. PAMI*, vol. 24, July, pp. 932–946.

Erol A., Bebis G., Nicolescu M., Boyle R. D., Twombly X., 2007. 'Vision-based hand pose estimation: A review'. *Computer Vision and Image Understanding* vol. 108, pp. 52–73

Gumpp T., Azad P., Welke K., Oztop E., Dillmann R., Cheng G., 2006. 'Unconstrained Real-time Markerless Hand Tracking for Humanoid Interaction'. *Proc. of 6th IEEE-RAS Intern. Conf. on Humanoid Robots*, pp. 88-93

Haiting Zhai, Xiaojuan Wu, Hui Han, 2005. 'Research of a Real-time Hand Tracking Algorithm'. *Proc. of ICNN&B 2005*

Heap T., Hogg D. , 1996. 'Towards 3D hand tracking using a deformable model', *Proc. of International Conference on Automatic Face and Gesture Recognition*, pp. 140-145

Isard M., Blake A., 1998. 'CONDENSATION - conditional density propagation for visual tracking', *Int. J. Computer Vision*, vol. 29, pp. 5-28

Isard M., Blake A., 1998. 'ICondensation: Unifying Low-Level and High-Level Tracking in a Stochastic Framework', *Proc. ECCV98*, pp. 5-28

Letessier J., Bèrard F., 2004. 'Visual tracking of bare fingers for interactive surfaces', *Procs. of UIST '04: 17th Annual ACM symposium on User Interface Software and Tec*

Liu N., Lovell B., Kootsookos P., 2003. 'Evaluation of hmm training algorithms for letter hand gesture recognition', *Proc. ISSPIT 2003*, Darmstadt, Germany

Liu Y., Jia Y., 2004. 'A Robust Hand Tracking and Gesture Recognition Method for Wearable Visual Interfaces and Its Applications', *Proc. of 3rd Int. Conf. on Image and Graphgics IEEE*

Mahmoudi F., Parviz M., 2006. 'Visual Hand Tracking Algorithms', *Geometric Modeling and Imaging--New Trends*, vol. 5/6, pp. 228 – 232

Oka K., Sato Y., Koike H., 2002. 'Real-time tracking of multiple fingertips and gesture recognition for augmented desk interface systems', *Proc of FGR '02*, p. 429.

Olafsdottir, H. et al. Is the thumb a fifth finger? A study of digit interaction during force production tasks. Exp Brain Res (2005) 160: 203–213

Open Computer Vision Library, [on-line], Available: http://sourceforge.net/ projects/opencvlibrary/

Pantrigo J.J., Montemayor A.S., Sanchez A., 2005. 'Local search particle filter applied to human-computer interaction', *Proc. of ISPA'05*, pp. 279 - 284

Shan Lu, Metaxas D., Samaras D., Oliensis J., 2003. 'Using multiple cues for hand tracking and model refinement', *Proc. IEEE Conf. on Computer Vision and Pattern Recognition 2003*, vol. 2, pp. 443-450.

Shan C., Wei Y., Tan T., Ojardias F., 2004. 'Real time hand tracking by combining particle filtering and mean shift', *Proc. FG2004*, pp. 669-674

Stenger B., Mendonca P.R.S., Cipolla R., 2001. 'Model-based 3D tracking of an articulated hand', *Proc. IEEE CVPR 2001* vol. 2, pp. II-310 - II-315

Stenger B., Thayananthan A., Torr P. H. S., Cipolla R., 2004. 'Hand pose estimation using hierarchical detection', *Proc. of Intl. Workshop on Human-Computer Interaction 2004*.

Tomasi C., Petrov S., Sastry A., 2003. '3D Tracking = Classification + Interpolation', *Ninth IEEE International Conference on Computer Vision*, vol, 2, pp. 1441–1448

Yang Liu, Yunde Jia, 2004. 'A robust hand tracking for gesture-based interaction of wearable computers', *Proc. ISWC 2004*, pp. 22 - 29

Yang Liu, Yunde Jia, 2004. 'A robust hand tracking and gesture recognition method for wearable visual interfaces and its applications', *Proc. International Conference on Image and Graphics ICIG 2004*, pp. 472 – 475