How to Make a Simple and Robust 3D Hand Tracking Device Using a Single Camera

ANDREA BOTTINO, ALDO LAURENTINI
Dipartimento di Automatica e Informatica
Politecnico di Torino
C.so Duca degli Abruzzi, 24, 10129 Torino
ITALY
http://www.polito.it/CGVG

Abstract: - In this paper we propose a non intrusive tracking system able to capture simple hand gestures in a fast and reliable way. Our device uses the images taken from a single camera to capture the 3D position and orientation of the hand of the user. In particular, the 3D position of the forefinger tip is used as a 3D marker, and the posture of the hand is used to input simple commands. The proposed approach combines several computer vision algorithms in order to exploit their strengths trying to minimize their drawbacks. The result is a real time system that is very robust against noise and cluttered backgrounds. An evaluation of the quality of the proposed approach is also presented.

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

1 Introduction

Hand tracking has recently received a great attention from the scientific community since it allows developing human computer interfaces which are more natural for the user. Gestures are expressive and meaningful body motions with the intent to convey information or interact with the environment. The expressive potentialities of an interface able to capture and interpret the gestures of the hand are extremely remarkable. The gesture interfaces are devices that measure in real time the position of the fingers (and sometimes of the wrist) of the user's hand in order to allow a natural interaction based on the recognition of the gestures. Such interfaces often require dedicated and expensive hardware. Furthermore, these devices are highly intrusive and they impose several constraints for their use. For instance the length of the cables connecting the device to the processing units restricts the movements of the user. Also, several sources of interferences can affect the precision of these tracking devices.

This paper presents a non intrusive hand tracking system which is capable to reconstruct 3D information from a monocular video sequence. The 3D position, orientation and posture of the hand are first reconstructed. From these data, a simple 3D mouse can be developed, using as 3D marker a reference point, the forefinger tip or the palm center when the hand is closed, and as commands a set of simple hand postures.

1.1 Related works

Several approaches have been presented in literature. They both differ for the features and for the models and algorithms used for tracking.

As for the features, color is the most commonly used. For hand tracking, the target color is skin, and the segmentation algorithms usually work on color spaces that separate illumination and chrominance values, like HSV space ([5], [7]), YUV [9] or YCrCb [8]. Several approaches ([10], [11]) integrate different visual cues, like motion and edges, into the segmentation process in order to make it more robust to varying illumination and cluttered backgrounds.

The modeling and tracking techniques proposed in literature are also different. The Mean shift algorithm [4] is a non parametric approach to detect the mode of a probability distribution using a recursive procedure that converges to the closest stationary point. However, the Mean shift algorithm cannot adapt to dynamically changing color distributions, like the one typically found in video sequences. The CAMShift algorithm, introduced in [5], is a generalization of the Mean shift algorithm taking into account this problem. Sequential Monte Carlo methods, like Condensation [6], Bayesian mixed state frameworks [12], particle filter based tracking algorithms ([13], [14]) have also been used. In [15] a model-based approach is proposed. The model of the moving object is created dynamically from the image sequence and then it is matched with the successive frames in the image sequence. More

complex models, like highly articulated 3D hand models [16] or deformable hand models [17], have also been proposed.

1.2 Our approach

In this work, we present a simple hand tracking system which reconstruct the posture of the hand and the position of a 3D reference point, which can be used as a 3D marker, from a single camera. The system works in real time. The approach combines several computer vision algorithms in order to exploit their strengths and to minimize their weakness. The hand silhouette is first extracted from the incoming image. Then a 3D model of the hand is fitted to the segmented image in order to reconstruct the hand posture and the position of the marker. The contribution of our work is a simple and computationally fast system, which is very robust against noise and cluttered backgrounds, and has also the advantage of being 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.

2 The hand tracking system

The reconstruction process is outlined in Fig. 1.

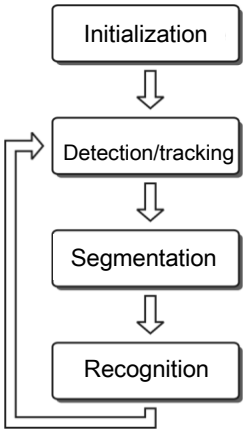

Fig. 1: outline of the reconstruction process

The main loop of the system 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 images
3. recognition: identifies the posture of the hand and the 3D marker position

These processes are detailed in the following subsections.

2.1 Initialization

Several initial parameters must be evaluated. First, the camera needs to be calibrated. The calibration process establishes a relation between the image plane and a fixed reference system in the 3D world. This is necessary, for instance, to project 3D points into the image plane. A copious literature is devoted to this problem. The most common approaches require, to evaluate camera parameters, a grid of 3D reference points, together with their corresponding 2D projections on the image planes, such as in Fig. 2. Efficient implementations of these techniques can be easily found over the Internet ([1], [2] and [3]).

Fig. 2: reference points on a calibration object

Second, the chromatic components of the objects to be tracked needs to be initialized since the tracking algorithm uses a probability distribution image of the desired color (that is, skin color in this case). This reference color is modeled with a chromaticity histogram created interactively by selecting a part of the image corresponding to the desired object. The RGB values of the pixels of the selected region are converted into the Hue Saturation Value (HSV) color system. HSV separates hue (color) from saturation (how concentrated the color is) and from brightness, and allows to create a simple color model taking 1D histograms from the hue channel.

2.2 Detection/tracking

The detection/tracking module is a state machine, whose state diagram is shown in Fig. 3. In the detection state, the input image is processed until a hand enters the image. Then the hand is tracked until it exits the image. The **`detect()`** function returns true when the object is detected, while the **`track()`** function returns false when the object is lost.

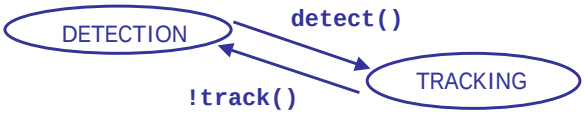

Fig. 3: state diagram of the detection/tracking module

This module uses two different algorithms:

- Mean shift for object detection
- CAMShift for object tracking

The input of the Mean shift 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. Basically, each pixel of the incoming image corresponds to a bin of the

reference histogram. The back projection is obtained by setting the pixel of the destination image to the value of this histogram bin.

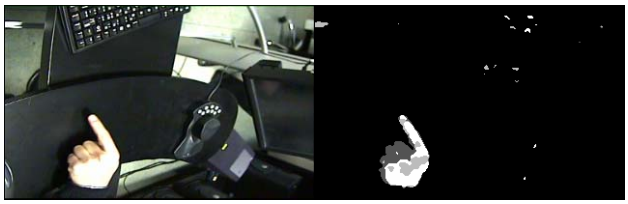

Fig. 4: an incoming image and the corresponding histogram back projection

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. 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. 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.

Summarizing, this module works as follows. Initially, Mean shift processes the incoming probability images until a hand is detected. Then, the position and dimension of the search window are used by CAMShift, which tracks the moving hand, computing for every frame the new position and dimension of the search window until the hand exits the image.

Before applying both algorithms, the probability image is filtered, as suggested in [5], in order to remove pixels whose brightness in the HSV image is too low since, where brightness is low (V near zero), saturation is also low (S near zero) and hue becomes noisy, because the small number of discrete hue pixels in these conditions cannot adequately represent slight changes in RGB.

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

2.3 Segmentation

The segmentation module, given position and dimension of R, extracts the silhouette of the hand from the probability image. We recall that the segmentation process is activated only if the object of interest has been detected.

The probability image is thresholded in order to obtain a binary image. Then, some morphological operators are applied (erosion and dilation), to fill holes in the silhouette and remove spurious pixels.

Fig. 5: a probability image and the extracted silhouette

Finally, the connected component contained in the search window is identified and its bounding box is evaluated. Any further processing on the images will take place on this Region of Interest (ROI), reducing the computational burden of the whole system. This allows also to discard other disturbing connected components not belonging to the hand.

The output of this module is a binary image containing the silhouette and its ROI.

2.4 Recognition

The recognition process is the most complex of the system. The input of this process is the silhouette of the moving object and the R and ROI regions. The output will be the 3D position of the marker and the posture of the hand.

A simple 3D model of the hand is used to reconstruct the desired information. The advantage of using a model based approach is that a 3D model has a well defined geometry that can be compared/fitted with the extracted silhouette and also that it allows to define a state that represents position, orientation and posture of the hand. Hence, the information obtained from segmentation is used to synthesize the model state.

The model is composed by one to three ellipsoid, 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 and the shapes of the corresponding models are shown in Fig. 6.

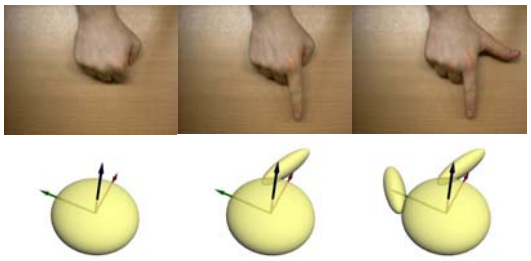

Fig. 6: the three hand postures identifiable and the corresponding model shapes

The projections of the ellipsoids on a plane are quadrics (Fig. 7) 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.

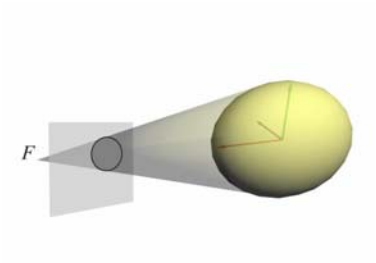

Fig. 7: projection of an ellipsoid on the image plane

The reconstruction process involves two steps:

- a reconstruction step, where the state of the model is reconstructed using the ICondensation algorithm
- a data filtering step, where a Kalman filter is used to smooth the noise in the extracted data

Icondensation [18] is a combination of the statistical technique of importance sampling with the Condensation algorithm [6]. 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 their weights are updated. From these weights, it is possible to predict the probability distribution over the search space at the next time instant. Thus, more probable states are more likely to be propagated over time and more than one probable state can be propagated at each time instant. The process is iterated for a predefined number of steps, and the finale 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, the result of the exor of model projection and silhouette of the hand (see Fig. 8), the weight of the sample *s* is given by:

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

Fig. 8: exor between the projection of the model and the silhouette of the hand

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 two information. The search window R given by the tracking step and the ROI corresponding to the bounding box of the identified silhouette (Fig. 9). 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.

Fig. 9: ROI, external rectangle, and R, inner rectangle

The first order moments of R give a reasonable indication of the center 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 idea 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.

The precise position of the 3D marker can then be identified. If the posture is 0 (closed hand), the 3D marker is given by the center of the palm. Otherwise, we identify the precise position of the fingertip on the image, projecting the palm on the image and using it to mask the region corresponding to the forefinger. The point of the major axis of the covariance ellipse of this region more distant from the center of the projection of the palm, is taken as the 2D position of the fingertip. The 3D marker

position is then given by the intersection of the line back projecting this point in 3D from the optical center of the camera and the principal plane of symmetry of the ellipsoid corresponding to the palm.

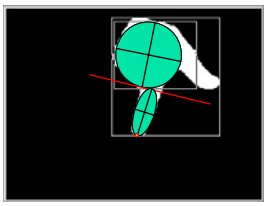

Fig. 10: precise identification of the forefinger tip

At the end of the reconstruction phase, in order to reduce the noise in the output data, a Kalman filter is applied to the position of the 3D marker.

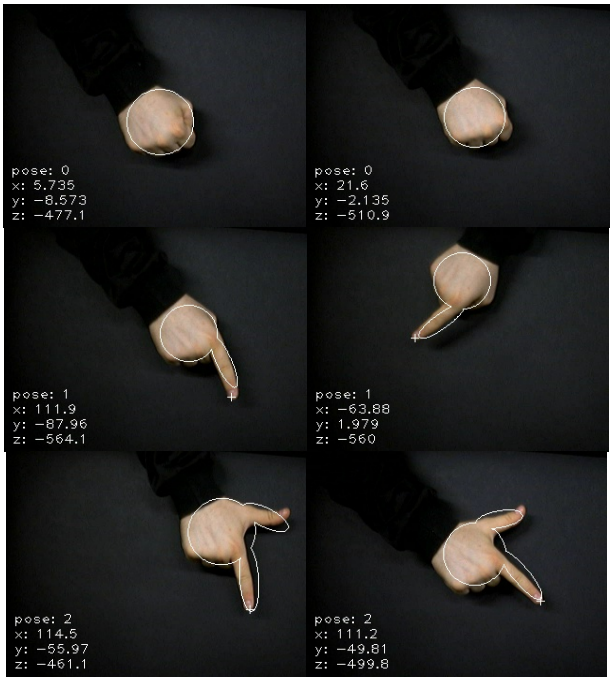

Fig. 11: tracking results

2.5 Examples

Some examples of the results obtained with the proposed tracker are shown in Fig. 10. The images show the projection of the 3D model on the incoming frames, together with an indication of the 3D marker position and of the detected posture. The position of the identified fingertip is also highlighted.

The tracker has been used to control a simple 3D environment. Pose 1 is used to move the marker position, while pose 2 mimics the click of the mouse. Dragging along X and Y axis of the image plane, that is moving the hand with pose 2 along X and Y, is used to rotate the 3D representation with a trackball paradigm. Moving the hand along the Z axis with pose 0 is used to zoom in and out into the 3D world. Pose 0 is used instead of pose 2 in order to avoid unexpected zooming effect when rotating the 3D world.

3 Evaluation of the System

The proposed technology can be evaluated at different levels. According to the literature on the subject and referring to the available commercial products, a set of desirable characteristics can be identified

3.1 Initialization and re-initialization

The system guarantees a robust initialization and re-initialization. CAMShift 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 along the whole image, allows easily to “hook” the object and then to identify its position. The fact that Mean shift works on a search window of fixed size is not a problem since in the next frame the system is in tracking state and CAMShift, which exploits a variable size search window, is used. Reconstruction errors due to this fact affect the process of one single frame, and therefore can be considered negligible.

3.2 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. As a matter of facts, chromatic components outside the range defined by the histogram of the color components we’re looking for, are cut away from the image by the histogram back-projection. An example can be seen in Fig. 4 and Fig. 5, where a complex background is present. Some small groups of pixels not belonging to the hand are present in the probability image, but most of them are discarded during segmentation. For skin-like objects entering the image, we stress that the CAMShift algorithm is very robust against distracters. Once CAMShift is locked onto the mode of a color distribution, it will tend to ignore other nearby but non-connected color distributions. Those blobs are then discarded, since any further processing will be applied only on the ROI region. It is true, however, that some problems can be caused when the disturbing object and the hand form a connected component.

3.3 Independence from illumination

The system is guaranteed to work for a wide range of variations of illumination intensity since the segmentation process is based entirely on the

chromatic components of the image, which are 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.

3.4 Computational manageability

The system has been tested on a computer with a CPU Intel DualCore E6600, 1GB Ram, which can be considered as a medium cost processing unit (the complete system, including capture board and camera has a cost lower than 1000€). Processing the image stream at a frame rate of 25 frames/sec, the mean latency is 20ms, and one single CPU is used at 30% of its capacity. 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.

3.5 Quality of the results

Accuracy and jitter of the tracker can be computed in the following way. Let's take a grid of points whose 3D position is known (for instance the corners of the calibration grid). The accuracy can be evaluated pointing the reference points and evaluating the differences of the output from the ground truth data. The resulting accuracy is 2.80 mm RMS. The jitter can be computed by placing the forefinger in a fixed position and computing the standard deviation of the reference position. The resulting jitter is 0.92 mm RMS. These results show that the tracker provides a sufficient precision for many applications.

4 Conclusions

In this paper we have presented a simple and robust system that is able to reconstruct in real time from a single view the 3D posture of the user's hand in real time. In particular, the 3D position of the forefinger is used as a 3D pointer, and three simple hand postures are used to mimic the functions of the mouse buttons. The 3D information is reconstructed from a single view. The problem appears to be unconstrained. However, some geometric properties of the hand silhouette extracted from the image can be used to reconstruct the full 3D posture of the hand within an acceptable precision. The experiments demonstrate that the proposed approach is very fast and robust. As a matter of fact, the tracking device can output data at a rate of 25 samples/sec, more than acceptable for building a simple 3D input device, and, on a low cost processing units, it uses only the 30% of the CPU's processing capabilities. This allows the concurrent execution of other applications that use the tracker data. The experiments also show that the accuracy, the jitter and the latency of the proposed system are more than acceptable for such a simple device.

References:

[1] Tsai Camera Calibration Code http://www-2.cs.cmu.edu/~rgw/TsaiCode.html

[2] Open Computer Vision Library, http://sourceforge.net/ projects/opencvlibrary/

[3] Camera Calibration Toolbox, http://www.vision. caltech.edu/ bouguetj/calib_doc/

[4] K. Fukunaga and L.D. Hostetler, "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition," IEEE Trans. Information Theory, vol. 21, pp. 32-40, 1975.

[5] Gary R. Bradski. "Computer Vision Face Tracking For Use in a Perceptual User Interface". Intel Technology Journal, Q2, p.15,1998.

[6] M. Isard and A. Blake, "CONDENSATION - conditional density propagation for visual tracking" Int. J. Computer Vision, 29, 5-28, 1998

[7] K. Imagawa, S. Lu, and S. Igi, "Color-Based Hand Tracking System for Sign Language Recognition," Proc. Int'l Conf. Automatic Face and Gesture Recognition, pp. 462-467, 1998.

[8] J. Dias, P. Nande, N. Barata, A. Correia "O.G.R.E. – Open Gestures Recognition Engine" Proceedings of the XVII Brazilian Symposium on Computer Graphics and Image Processing IEEE 2004

[9] N. Liu, B. Lovell, and P. Kootsookos. "Evaluation of hmm training algorithms for letter hand gesture recognition". Proc. ISSPIT, December 2003.

[10] C.-L. Huang and W.-Y. Huang, "Sign Language Recognition Using Model-Based Tracking and a 3D Hopfield Neural Network," Machine Vision and Application, vol. 10, pp. 292-307, 1998.

[11] Shan Lu, D. Metaxas, D. Samaras and J. Oliensis, "Using multiple cues for hand tracking and model refinement", IEEE Conf. on Computer Vision and Pattern Recognition 2003, 18-20 Jun. 2003, vol.2, pp. 443-450.

[12] M.Isard and A.Blake, "A mixed-state condensation tracker with automatic model-switching", ICCV98, Jan. 1998, pp. 107- 112

[13] Wen-Yan Chang; Chu-Song Chen; Yi-Ping Hung, "Appearance-guided particle filtering for articulated hand tracking", Proceedings of CVPR 2005. Pag. 235 - 242

[14] C. Shan, Y. Wei, T. Tan, F. Ojardias, "Real time hand tracking by combining particle filtering and mean shift", Proc. FG2004, Pag. 669- 674

[15] C.-L. Huang and S.-H. Jeng, "A Model-Based Hand Gesture Recognition System," Machine Vision and Application, vol. 12, no. 5, pp. 243-258, 2001

[16] J. M. Rehg and T. Kanade. "Visual tracking of high DOF articulated structures: an application to human hand tracking". Proc. 3rd European Conf. on Computer Vision, pages 35–46. 1994

[17] A. J. Heap and D. C. Hogg. "Towards 3-D hand tracking using a deformable model". Proc. 2nd Face and Gesture Recognition Conf., pages 140–145, 1996.

[18] Y. Liu, Y. Jia "A Robust Hand Tracking and Gesture Recognition Method for Wearable Visual Interfaces and Its Applications" Proceeding of the Third International Conference on Image and Graphgics 2004