Dynamic Time Warping for Binocular Hand Tracking and Reconstruction

Javier Romero, Danica Kragic
CAS-CVAP-CSC
KTH, Stockholm, Sweden

Ville Kyrki
Dept. of Information Technology
LUT, Lappeenranta, Finland

Antonis Argyros
Institute of Computer Science
FORTH, Crete, Greece

***Abstract*— We show how matching and reconstruction of contour points can be performed using Dynamic Time Warping (DTW) for the purpose of 3D hand contour tracking. We evaluate the performance of the proposed algorithm in object manipulation activities and perform comparison with the Iterative Closest Point (ICP) method.**

I. Introduction

In a learning by demonstration context, a robot observes a human performing a task, after which it is supposed to perform the action and thus learn through an imitation process. In order to imitate a human action, the robot needs to retrieve information about of how a specific task was performed – it needs to register the movement of the whole or parts of a human body and an object, as well as the sequence of different actions performed on the object. The goal of our current work is the development a real-time stereo based hand tracking system that can be used for 3D hand pose estimation required in the imitation process.

Tracking and reconstructing hands in 3D requires solutions to a number of different problems: hand modeling and detection, temporal association, representation and extraction of 2D data, data association, and matching for 3D reconstruction. Most of the approaches of 3D hand tracking are either model-based or view-based. The former rely on articulated 3D hand models, used to minimize an error function between the model and the observed image data in a sequence of images. This approach requires a model initialization in the first frame which is commonly performed manually. View-based approaches perform pose estimation and classification using a limited number of selected hand poses collected in the training stage. Related to the number of cameras used, both monocular and stereo systems have been used. In the case of the former, assumptions about the size of the hand have to be made to facilitate the 3D reconstruction problem. The stereo-based approaches, on the other hand, require a data association and matching step prior to the reconstruction.

This work is based on further developments of the system presented in our previous work [2]. In the tracking system, the hands are first identified separately in each of the stereo images and their contours are extracted. This is followed by stereo-based blob matching technique and shape matching through contour alignment. The particular objective of the work presented here is the development and evaluation of the shape matching and contour alignment step. In the original work [2], Iterative Closest Point (ICP) algorithm and an assumption of the affine motion model were used. However, this approach is not suitable for cases where the inherent assumption of planarity of the hand due is not valid which is commonly the case in object grasping and manipulation activities.

This paper presents a new approach for contour alignment based on dynamic programming, considered here in the Dynamic Time Warping (DTW) context. An extensive experimental evaluation shows that the performance of the new approach is clearly superior, increasing the robustness to occlusions and relaxing the planarity assumptions.

II. Related work

Pose tracking and 3D reconstruction of hands is a difficult problem: hands are textureless objects with many degrees of freedom, usually self-occluded or occluded by other objects when object manipulation actions are considered. Since full 3D reconstruction based only on the hand depth estimation is a complex and time-consuming process, view-based approaches have been extensively used in the literature [3]. The tracking problem is then solved through a classification framework, relating image information directly to the pose space of the hand. Approaches that make use of databases and deformable templates fall in this group [5], [6]. Model-based approaches [3] commonly build an articulated hand model. This model is then used in the tracking process where the incremental change in pose between consecutive images is estimated by minimizing an error function between the model and the observed image data.

For robotic imitation scenarios, where it is expected that a human demonstrates to a robot how to manipulate a certain object in its workspace, model-based approaches offer a better solution. Apart from not having the need for extensive training, database generation and storage, model-based methods offer a more flexible framework once the mapping between different kinematic chains is needed.

Even if a considerable amount of work has been put on the development of humanoid robots during the past few years, robot hands are still simple and do not offer the full complexity of human hands. The simplest form of mapping from a human to a robot hand may then be to just disregard those degrees of freedom that are not articulated on the robot hand. Model-based trackers also offer the capability of continuous pose estimation while view-based methods, if they are not extended with some local fitting step, only provide classification to the nearest dictionary pose.

Within the model-based approaches, we can differentiate systems based on the extracted features and the methods used to reconstruct the hand model based on these features. Regarding the features, we can differentiate between low level and high level (semantic) features. High level features are desirable since they compress a lot of information about the hand pose in few parameters, and they allow high processing speed for the fitting process. The drawback of high level features is that it is difficult to extract them from images in a general and robust way. One of the most common examples of a high level feature is the position of the fingertips [4].

Regarding the number of cameras, there are some important differences between monocular and multi-camera systems. While the monocular systems usually use some predefined hand parameters, such as the length of phalanxes and the size of the palm, to reconstruct the hand in 3D [4], stereo systems can extract depth information from the image data directly, without assumptions about the hand parameters. The extraction of depth information can be done in many different ways. There are approaches such as [1] where correlation methods are applied to extract dense depth maps of the hand. However, the lack of texture usually makes the use of correlation matching difficult. In our previous work, [2] a different approach was applied considering only the reconstruction of the hand contour.

III. Contour Matching

In this section, we briefly introduce the ICP and DTW algorithms.

A. ICP

ICP algorithm computes a motion which transforms one set of points into another one according to a model. The algorithm is iterative: At each iteration, it first computes a motion which minimizes the current matching error and then applies the estimated motion to update the point correspondences. Two strong assumptions are made:

- An initial approximation for the point correspondences is available.
- A suitable motion model is available to perform the contour alignment.

In [2], the initial error measure is based on the first and second moments of the contours. The hand contours are approximated by ellipses and their centroids and principal axes are matched. This approximation is fast, but it has some problems: the more circular this ellipse approximation of the contour is, the more uncertain is the alignment of the axes. The motion model chosen in [2] is an affine transformation. In terms of stereo matching, this works fine as long as the contour points lie in a plane which also restricts the algorithm to work only for a very limited set of hand poses. Finally, ICP needs a measure of the error during the matching process. In the original work, the 2D squared distance between a point and the transformed point was used.

B. DTW

The approach adopted here, dynamic time warping, is conceptually quite different. It is not iterative, and it computes the point correspondences without any assumption of the underlying motion model. The algorithm consists of four steps:

1) Compute the pairwise distances from each point in one set to all the points in the other set.
2) Select a pair of points that are supposed to be a good match, in order to initiate the matching process.
3) For each possible pair of points, compute the accumulated cost of reaching this pair, based on the accumulated cost of previous points and the cost of the jump from the previous pair (the pair with minimum accumulated cost previously computed).
4) The optimal path corresponding to the minimum total cost of matches can be extracted by tracing back from the end point.

The algorithm has time complexity $O(n^2)$ where n is the number of points to be matched.

In the remaining part of the section, we present the principal details of the proposed approach: the building of the distance matrix and the choice of the starting point.

1) Distance Matrix: DTW algorithm finds the set of correspondences with the least total cost (or distance) between the matched points. For this reason, it is very important to choose a distance measure that in a good way represents the similarity of two hand contours in a stereo pair of images.

From a geometric point of view, the relation between points in a calibrated stereo pair is given by the essential matrix E:

$$P_R^{\mathrm{T}} E P_L = 0 \tag{1}$$

The relation is applied to points P_R and P_L in the normalized camera coordinate systems, and are also valid for their normalized image plane projections p_R and p_L. Considering the camera intrinsic parameters K^{-1}, we obtain the well known fundamental geometry relationship:

$$p_L = K_L^{-1}\overline{p}_L \qquad p_R = K_R^{-1}\overline{p}_R \tag{2}$$

$$p_R^{\mathrm{T}} E p_L = 0 \tag{3}$$

$$\overline{p}_R^{\mathrm{T}} K_R^{-\mathrm{T}} E K_L^{-1} \overline{p}_L = 0 \tag{4}$$

$$\overline{p}_R^{\mathrm{T}} F \overline{p}_L = 0 \tag{5}$$

$$\overline{u}_R = F\overline{p}_L \tag{6}$$

$\overline{p}_R$ and $\overline{p}_L$ represent pixel coordinates and $\overline{u}_R$ represents the epipolar line.

The distance from a point to the epipolar line generated by another point can be used in order to build the distance matrix. The main advantage of this measure compared to other measures such as the coordinates of the points or the coordinates with the origin in the hand centroid is that the measure is not based on any assumption except the epipolar geometry: if the camera calibration is correct (and there are no occlusions), the corresponding point lies on the epipolar line. The main disadvantage is that this measure might be ambiguous: in the case of point denoted 0 in Figure 1, it

will perfectly match both points denoted 0 and 8 in another image.

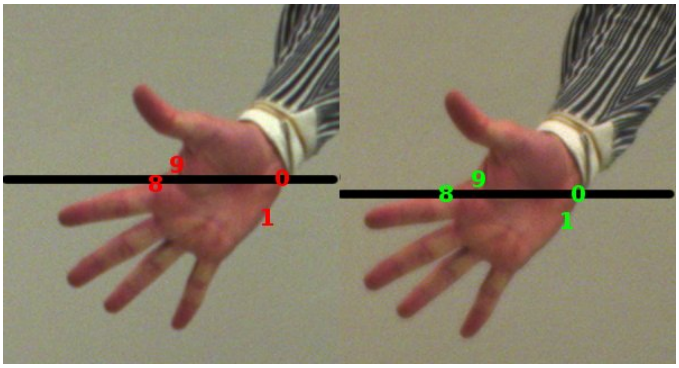

Fig. 1. Point 8 has distance 0 to point 0 since it is based on the distance between the epipolar lines the points lie on.

However, this problem is solved by the DTW algorithm. Although two points would have similar distances to each other, the subsequent ones will probably have very different distances (as points 1 and 9 in Figure 1). This means that despite the distance in the first pair will be low, the following matches will increase a lot the total cost of the matching, rejecting finally this set of correspondences. When there are points with similar distances close to each other, the system will produce errorneous results. This can be seen in Figure 12, where the wrist segment of the contour is wrongly reconstructed since it is almost parallel to epipolar lines.

Another problem solved by the DTW algorithm are the occlusions. If some points are occluded just in one of the images of the stereo pair, the number of contour points will be different in each image. When the algorithm reaches the point where the partial occlusion begins, it detects that the distance between the next pairs increases until the occlusion finishes. If a suitable distance measure is chosen, DTW will drop these points, that is, it will leave them without any correspondence in the other image. The matching will continue normally when contour points are visible in both images again, see Figure 2.

2) Starting Point Selection: Matching cyclic sets of points with DTW has one prerequisite: the beginning and end matching pair have to be chosen and it has to be the same point pair. Although DTW can align sequences with disaligned starting points, the points that are dropped while aligning the sets are lost. In Figure 3 we show an example of that: the real starting point pair should be the pair $(5, 0)$ instead of $(0, 0)$. As the consequence, only the central point pairs of the contours, from $(5, 0)$ to $(14, 9)$, are well matched. There are solutions to this problem, but they usually require at least twice the computation time required by the original DTW algorithm.

It seems reasonable to use the distance matrix built with distances to epipolar lines in order to use the point pairs with lowest distance as starting points. Unfortunately, as said, this measure is ambiguous, so we have many pairs with very low distances which do not represent good matches. One

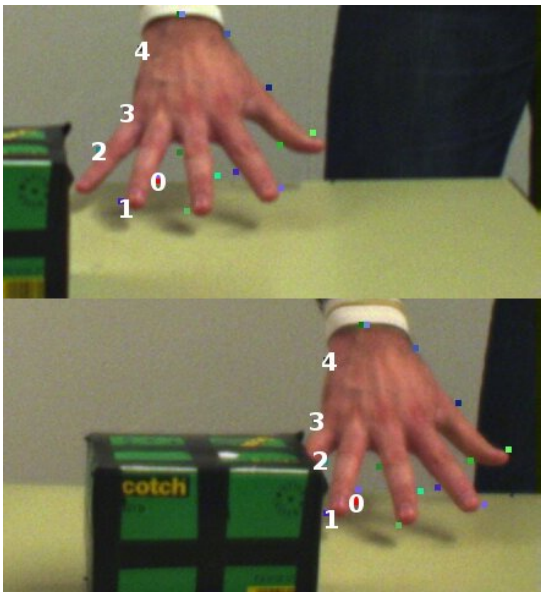

Fig. 2. Only point 2 is wrongly matched due to the occlusion.

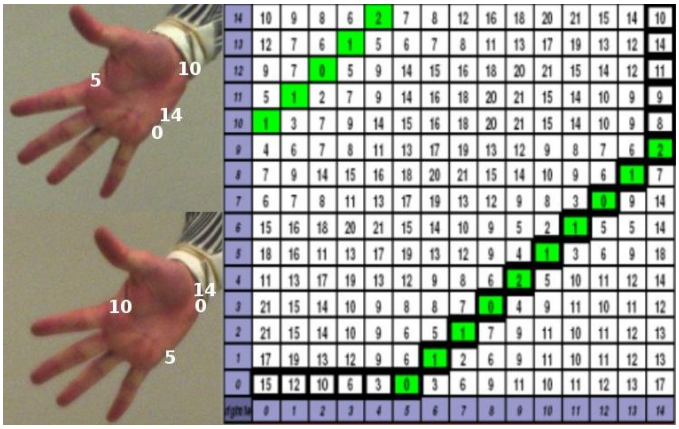

Fig. 3. Gray cells corresponds to the ideal correspondences, and bordered are the real ones.

can think about applying another measures such as using the coordinates centered in the middle of the hand to fix this ambiguity. However, there is another problem: since the distance to an epipolar line is a continuous real value, a threshold should be applied to consider a point "close" or "far" to the epipolar line and it may be difficult to find such a threshold.

For this reason a different approach was taken. If we assume that the hand pose does not change too much between the consecutive frames, a good match from the current frame after applying the DTW algorithm, which avoids the ambiguity of the distance measure, can be used as a starting point. In the first frame, the starting point is selected based on the distance matrix. Although the matching can be wrong in some points, there is a region between the alignment portion in the beginning and the end, see Figure 3, where the points are well matched. With this procedure, the selection of starting point is fast and accurate and the problems may

occur only in the first frame.

C. Accumulated distance computation

Once the starting point pair has been selected, we can begin the computation of the accumulated cost until each possible point pair in the matrix has been visited. The allowed transitions and their related costs J have to be defined. There are different possibilities related to the allowed transitions in a DTW system. We describe below what each of the possible transitions means:

- From pair (m, n) to pair $(m + 1, n)$: we advance one point in the first contour but stay in the original point on the second contour. We denote this an "alignment" jump.
- From pair (m, n) to pair $(m + 1, n + 1)$: we advance one point on both contours. We denote this a "matching" jump.
- From pair (m, n) to pair $(m + 2, n)$: we advance two points in the first contour but stay in the original point on the second contour.
- From pair (m, n) to pair $(m + 2, n + 2)$: we advance two points on both contours.

In our system, we only allow the transitions from (m, n) to $(m, n + 1), (m + 1, n)$ (alignment) and $(m + 1, n + 1)$ (matching). We have tested the performance of the system by allowing longer transitions and we concluded that the improvement was not significant.

In the proposed approach, the cost associated to a transition serves as a multiplier of the distance associated with a point pair. It can be used to favor shorter transitions, or to favor "matching" transitions, for example. For example, if there are no occlusions, we would want to favor matching transitions where we advance one point in both contours, and not transitions for alignment, where only one contour advances for one point. However, we experienced that the behavior with this approach is worse in cases of partial occlusions, where a lot of alignment transitions are required to match the contours properly. A good balance in our system was a factor of 1.5 for alignment, and a factor of 1 for matching, meaning that alignment has an additional cost. This improved considerably the matching process in the fingertips, where there are points with similar distances very close.

The accumulated cost C for a pair (i, j) if the last matched point was (m, n) can then be expressed as:

$$C(i, j) = C(m, n) + \quad (7)$$
$$min_{allowed\ jumps}(J(m - i, n - j) \times c(i, j))$$

An example with constant cost for the different transitions J is shown in Figure 4. In order to compute the accumulated cost in the white bordered cell $(4, 4)$, we add the intrinsic cost of the pair $c(4, 4) = 1$ and the minimum accumulated cost for the possible predecessors, that in this case is an alignment transition from $(4, 3)$. J is set to one in this case for simplicity. This accumulated cost is calculated for all the point pairs until the end point pair is reached.

1	5 | 26	7 | 22	12	1
4	13 | 21	9 | 15	1 | (1+3)	17
3	8 | 8	5 | 6	2 | 3	15 | 18
2	0 | 0	1 | 1	7 | 8	11|19
right/left	2	3	4	1

Fig. 4. Distance matrix with accumulated costs.

D. Backtracking

Once all the accumulated distances have been calculated, the backtracking process begins. The set of best correspondences is extracted by backtracking the "best predecessor" from the end point pair and repeating the process until the starting point pair is reached.

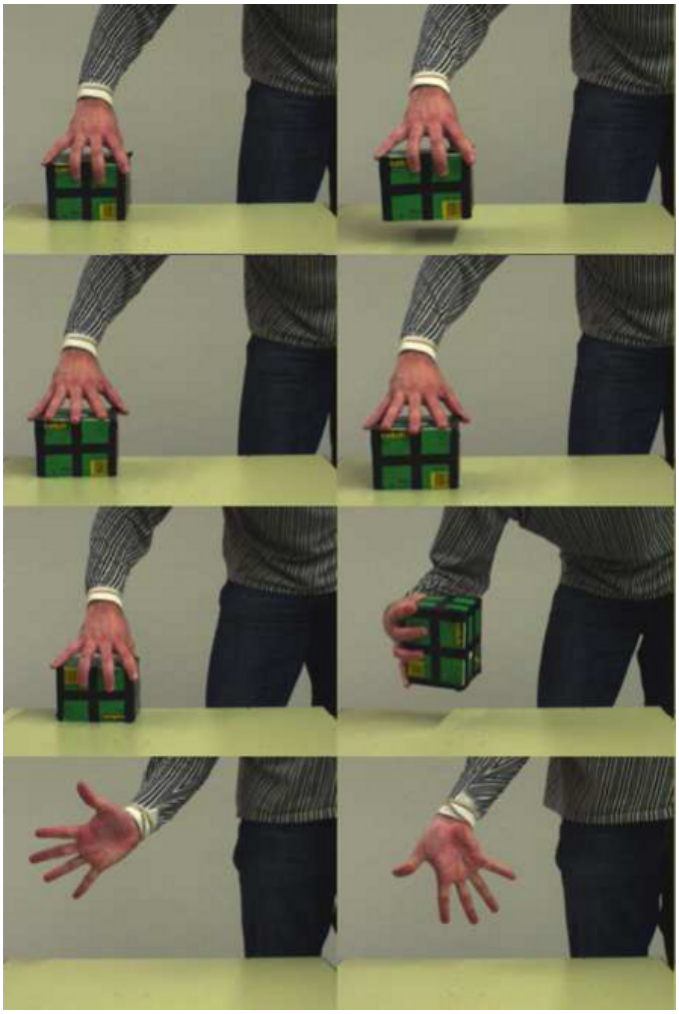

Fig. 5. Two frames of each sequence used in evaluation: moving, pushing, rotating an object and simple hand waving.

IV. Experimental evaluation

We compared the performance of the ICP algorithm developed in [2] and the DTW algorithm developed proposed here in a number of hand gesture and object manipulation sequences. For the ICP algorithm, we allow the maximum

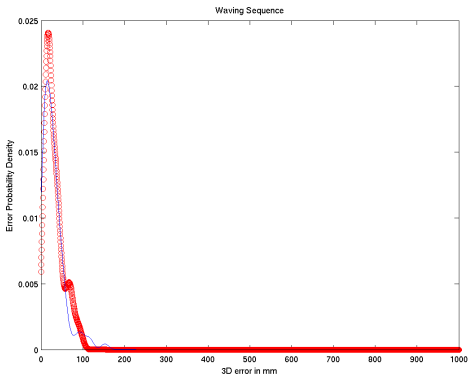

Fig. 6. Waving sequence: ICP(thick/circle line) and DTW (thin line).

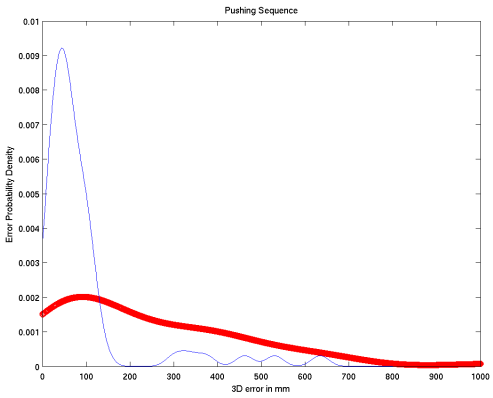

Fig. 7. Pushing sequence: ICP(thick/circle line) and DTW (thin line).

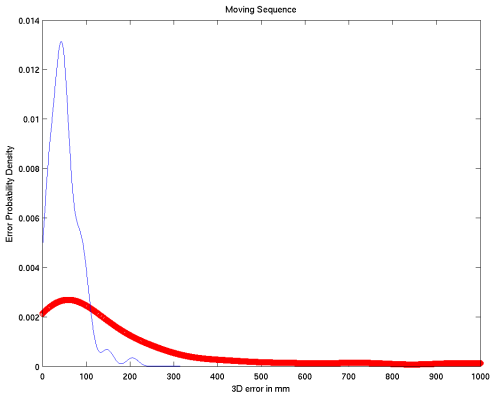

Fig. 8. Moving sequence: ICP(thick/circle line) and DTW (thin line).

number of iterations to be 50. The sequences on which the performance of the algorithms was evaluated included moving, pushing and rotating an object, and simple hand waving, see Figure 5. Ground truth was provided by manually marking the matched pairs in all frames. The results are represented by plotting the error probability density function.

The waving sequence is the simplest one. The hand contour points lie in the same plane, facing the cameras and without any object involved. These are the ideal conditions

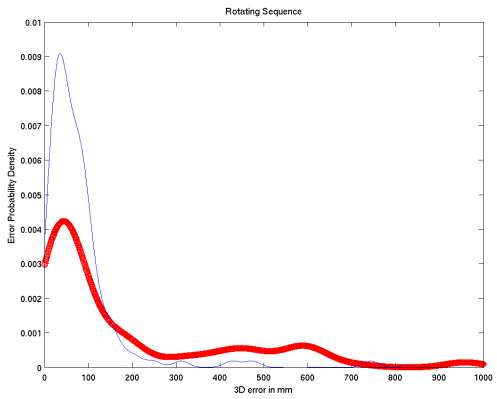

Fig. 9. Rotating sequence: ICP(thick/circle line) and DTW (thin line); Rotating sequence

of the ICP algorithm, and the performance of this algorithm in this sequence is very good. The performance of DTW in this sequence is slightly worse, see Figure 6.

The rest of the sequences consist on the manipulation of an object, so there are occlusions. ICP is less robust to occlusions than DTW, since usually the hand contour shape is considerably different in stereo images when there are occlusions present, see Figure 2. But the main advantage of DTW is that it shows a good performance for the case when the hand contour points do not lie on a plane, see Figures 7, 8 and 9. Those figures represent the probability density of the distance between the extracted fingertips and the ground truth. As we can see, the distance error in DTW is concentrated in lower values than in ICP.

We have also performed a qualitative evaluation of the two methods. Some of the sample results are presented in Figures 10-12. Here, the reconstruction of hand contours performed with DTW and ICP is compared to the ground truth which is represented as a line skeleton from the wrist to the fingertips. In Figure 10, it is visible that the performance of DTW is better than ICP when the planar assumption is not satisfied anymore. Even if the extracted contours are almost the same in the image, the reconstructed 3D contour is clearly much better in the case of DTW approach. Figures 12 and 11 also show that the DTW approach not only outperforms ICP but also performs well in cases of occlusions. While the thumb is occluded by the rest of the fingers, it is partially visible in the 3D reconstruction since the right camera image (not present in the figure) had a better angle to visualize the thumb.

V. Conclusions

Extraction and tracking of human hands is an important part of various interaction and instruction systems. Many systems have been proposed in literature, based both on single and multiple cameras. However, most of them are designed for a specific purpose such as extraction of the hand contour without the full reconstruction of the hand's pose. The systems that can extract full pose of the hand are

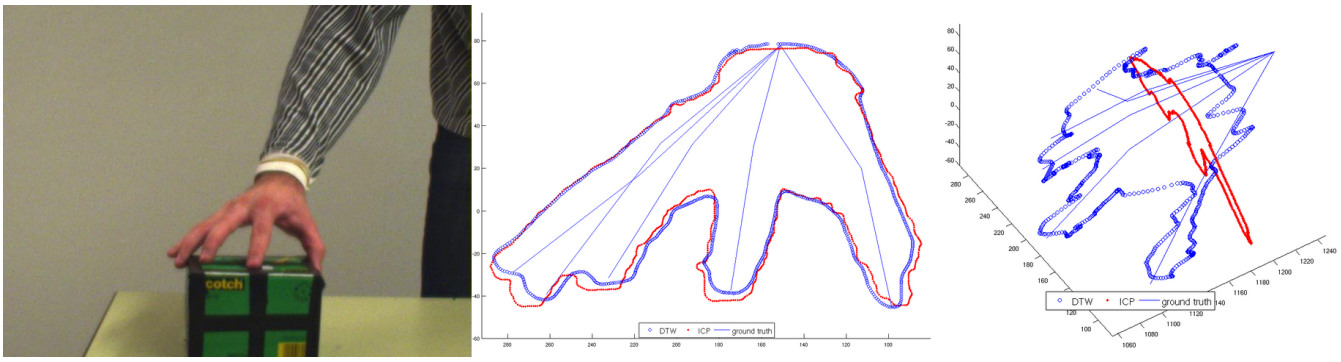

Fig. 10. ICP, DTW and ground truth reconstruction for the pushing sequence.

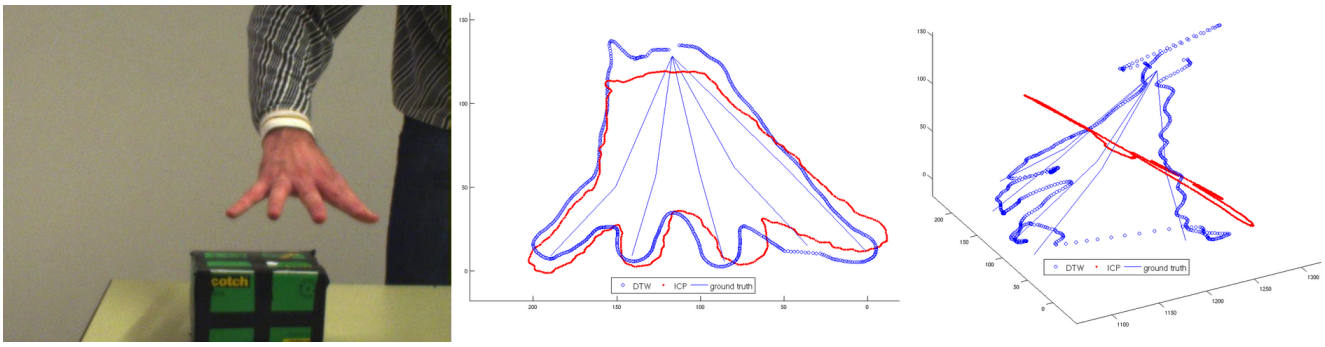

Fig. 11. ICP, DTW and ground truth reconstruction for the moving sequence.

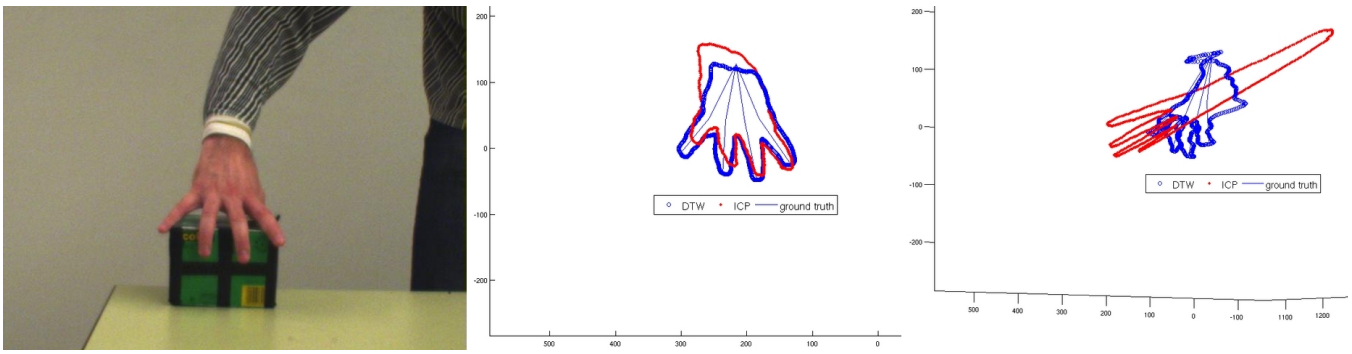

Fig. 12. ICP, DTW and ground truth reconstruction for the rotating sequence.

mostly run off-line or require parallel processing on several machines to achieve real-time performance.

Our current work aims at developing a full hand pose tracking system that performs in real-time without any special hardware. For this purpose, we use a stereo setup and built upon our previous work on hand contour tracking that assumed only cases where the hand was kept planar. To allow for more complex cases of object manipulation, we propose to replace the original ICP algorithm with a DTW approach that clearly shows a better performance in the considered sequences. Based on the contour extraction and fingertip detection in 2D, followed by the 3D matching and reconstruction step, the system will be used together with an articulated model of the hand to estimate the state of all the joints of the hand with the use of inverse kinematics.

VI. ACKNOWLEDGMENTS

This research is supported by EC project PACO-PLUS, FP6-2004-IST-4-27657 and the Swedish Research Council.

REFERENCES

[1] Q. Delamarre and O. Faugeras, Finding Pose of Hand in Video Images: A Stereo-Based Approach, *FGR*, 1998, pp 585-590.

[2] A. A. Argyros and M. I. A. Lourakis, Binocular Hand Tracking and Reconstruction Based on 2D Shape Matching, *Proc. of the IEEE International Conference on Pattern Recognition*, 2006, pp.207-210.

[3] V. Pavlovic, R. Sharma, T. S. Huang, Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review", *IEEE Trans. Pattern Analysis Machine Intelligence*, 1997, pp 677-695

[4] J. M. Rehg and T. Kanade, DigitEyes: Vision-Based Hand Tracking for Human-Computer Interaction, *CMU-CS*, 1993

[5] P. R. S. Mendonça and R. Cipolla, Model-Based 3D Tracking of an Articulated Hand, B. Stenger and *CVPR*, 2001, pp 310-315

[6] C. Tomasi, S. Petrov, A. Sastry, 3D Tracking = Classification + Interpolation, *ICCV*, 2003, pp 1441-1448