Recognizing and Tracking Human Action

Josephine Sullivan and Stefan Carlsson

Numerical Analysis and Computing Science, Royal Institute of Technology, (KTH), S-100 44
Stockholm, Sweden. {sullivan,stefanc}@nada.kth.se

Abstract. Human activity can be described as a sequence of 3D body postures. The traditional approach to recognition and 3D reconstruction of human activity has been to track motion in 3D, mainly using advanced geometric and dynamic models. In this paper we reverse this process. View based activity recognition serves as an input to a human body location tracker with the ultimate goal of 3D reanimation in mind. We demonstrate that specific human actions can be detected from single frame postures in a video sequence. By recognizing the image of a person's posture as corresponding to a particular key frame from a set of stored key frames, it is possible to map body locations from the key frames to actual frames. This is achieved using a shape matching algorithm based on qualitative similarity that computes point to point correspondence between shapes, together with information about appearance. As the mapping is from fixed key frames, our tracking does not suffer from the problem of having to reinitialise when it gets lost. It is effectively a closed loop. We present experimental results both for recognition and tracking for a sequence of a tennis player.

Keywords: human motion, tracking, shape, correspondence

1 Introduction

Two of the most important applications of the analysis of human motion in image sequences are 3D reconstruction and action recognition. Using dynamic models of interrelated shape primitives is traditionally the dominant approach to these problems often resulting in very impressive demonstrations [4, 8, 15, 9, 11, 16–18]. Successful 3D dynamic human modelling would simplify the task of inferring the underlying human activity. The main drawback of this approach, however, is that the tracking is not performed in a closed loop. Once the tracking fails, it has to be manually reinitialised. Automatic initialisation of a model based tracker requires the recognition of the 3D pose of the person being tracked. Recognizing the pose of a person is very often equivalent to recognizing the action taking place. If the main purpose of the 3D dynamic modelling is to recognize actions, this creates a paradox. Automating 3D model based tracking requires solving the problem for which the tracking was devised: action recognition. This applies, of course, only if we consider human pose and activity recognition as being equivalent. This remains to be proved. However, the problems of tracking, 3D modelling and recognition seem far more interrelated than has so far been considered. Introducing pose recognition into the tracking paradigm makes it into a closed loop system and one capable of automatic recovery from failure modes. However, if we have a system for 3D pose detection it could equally well be used permanently during the tracking, not just for error recovery.

The idea that recognition and stored prior information should precede tracking, or even totally replace it, has been expressed explicitly in recent works. Howe et. al [12] use manual initialisation and subsequently rely on prior learned 3D information, combined with image tracking, to capture the motion of body parts. Brand [3] uses shadows and silhouettes to recognize complete 3D motion sequences. Toyama and Blake [19] use the idea of key frame recognition. This totally replaces the goal of tracking specific body parts. In contrast to these, our aim is to show that recognition of specific key frames, similar to that in [19] can be used in order to aid the frame to frame tracking and to close the tracking loop. The idea is to store a set of key frames to represent a specific action. These key frames are matched to the frames of an actual image sequence using an algorithm for computing qualitative shape equivalence [5]. This algorithm produces point to point correspondence between the key frame and the actual frame. Using this correspondence field we can transfer any body location on the key frame to that of the actual frame. Point transfer using the shape matching can, of course, also be computed from frame to frame in the actual sequence. This, however, is still a standard open loop tracker. The use of the key frames for point transfer closes the tracking loop and allows for error recovery at any instant, provided there is a key frame similar enough to the actual frame to allow for matching.

The paper focuses on three distinct but interrelated topics. Initially the shape matching algorithm is described. It is an improved version of the algorithm in [5] using combinatorial geometric hashing. Its power is demonstrated by applying it to the problem of action recognition, in our case this equates to the detection of action-specific poses. In particular the beginning of forehand strokes are extracted from a long sequence of tennis footage. At this stage we now know that it is possible to locate image frames similar to specific key frames. Our tracking paradigm is then explained. It is based upon transferring body locations from appropriately matched key frames. The last part of the paper describes how to achieve robust point transferral in the presence of imperfect data.

2 Shape Correspondence by Combinatorial Geometric Hashing

If we subjectively consider two images to be similar as in figure 1 we are almost always able to map a certain point in one image to a specific point in the other. This ability to define a correspondence field between the two images can be taken as a starting point for defining equivalence between shapes. If the field represents a smooth deformation of one shape into the other we are likely to consider the shapes as similar or belonging to the same category. The smaller the deformation the larger the similarity. Computing a correspondence field between two shapes enables us to measure the similarity of the shapes without any prior segmentation. The process of computing correspondence relies on the ability to define invariants i.e properties common to the two shapes. Since correspondence is between points on the shapes, these invariants should be computed from local pointwise information. For well defined transformations such as rigid or linear, invariants can be computed by simple algebraic manipulations. For general smooth deformations however, invariants are associated with qualitative topological shape properties. In order to compute correspondence fields between shapes that are smooth deformations of each other we are faced with the problem of computing qualitative invariants from sets of points on the shapes.

A certain point on a shape has a location and in general also a tangent direction. A complex of points and lines is formed by sampled shape points and their associated tangent lines. Consider these complexes from two shapes when the points are in "perceptual correspondence". Note that a certain line in one complex intersects between two points in exactly the same way in the two shapes. The order of the points $a_1 \ldots a_4$, $b_1 \ldots b_4$ is preserved as they are traversed e.g clockwise and so is the order of the line directions.

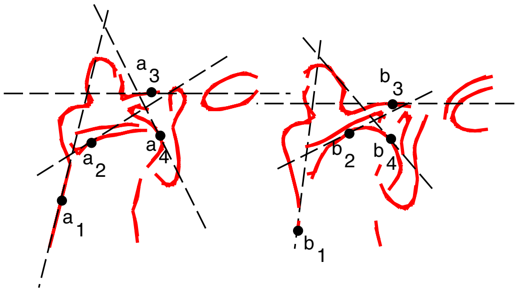
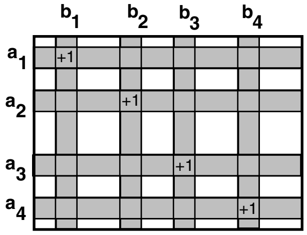


Fig. 1: Two point-line sets with the same topological type.



	b_1	b_2	b_3	b_4
a_1	+1			
a_2		+1		
a_3			+1	
a_4				+1

Fig. 2: A voting matrix with rows and columns corresponding to the points on the two shapes is updated whenever two 4-point-line complexes have the same topological type. The exact scenario is displayed for the example in figure 1.

The three components:

> point order, line direction order, and relative intersection of the lines and the points in a complex

define the *topological type* of the point-line complex. By defining the leftmost point to be the first point, we get a canonical ordering of the points.

It should be noted that this definition of invariants actually defines the class of smooth transformations that relate equivalent shapes. This class will contain any linear transformation of shapes that preserves orientation of points and lines, i.e. any linear transformation associated with viewpoint changes and planar shapes.

Point to point correspondence between two shapes can now be computed in the following way:

- Points are sampled equidistantly on the shapes A and B and every four point combination $a_1 \ldots a_4$, $b_1 \ldots b_4$ is selected.

- Whenever two sets of 4-point combinations have the same topological type, the matching table is updated by one vote for the correspondences $a_1 \leftrightarrow b_1 \ldots a_4 \leftrightarrow b_4$, figure 2.

- Unique correspondences are computed by applying the greedy algorithm to the final matching table:
  1. The maximum entry of the voting table is found and correspondence is declared between the points in the shapes A and B representing this entry.
  2. The row and column of this entry are deleted from the table.
  3. Repeat from 1.

This is essentially a discrete geometry version of the geometric hashing algorithm of [13] where topological type index replaces the quantized affine coordinates of the fourth point in the basis of the first three. By basing the index on qualitative geometric properties the bins of geometric hashing are defined generically and we avoid the sometimes arbitrary choice of defining the bins in based on metric or affine coordinates.

Figure 3 shows an example correspondence field found after applying this algorithm. Computing the voting matrix, in geometric hashing based shape correspondence,

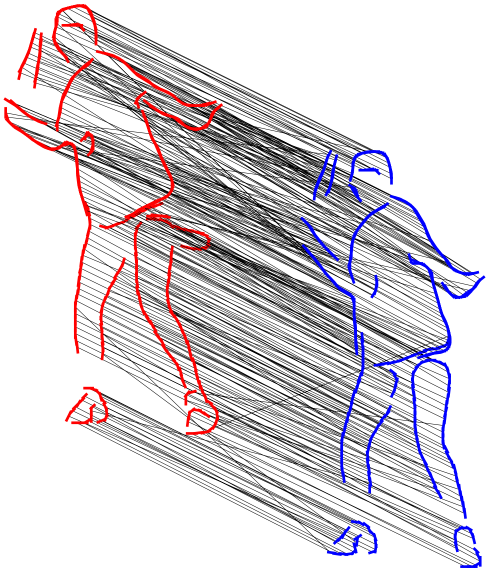


Fig. 3: The full set of correspondences computed between two frames.

can be formulated very efficiently as a matrix product [6]. The rows in these matrices actually correspond to the “shape context” vectors of [1] which have been recently applied to shape matching with considerable success.

3 Key Frame Based Action Recognition

The correspondence field that is computed from the shape matching algorithm contains all the information about the relations between image A and image B. If A and B are images with a well defined deformation between them, ideally this will be captured by the correspondence field. Let $p_1^a \ldots p_n^a$ and $p_1^b \ldots p_n^b$ be the coordinates of the corresponding points in image A and B respectively and $\mathbf{T}$ the class of transformations that, we know a priori, defines the relation between A and B. The decision that A and B are related by a member of this class of transformations can be based on the magnitude of the residual:

$$\min_{T \in \mathbf{T}} \sum_{i=1}^{n} ||p_i^b \ - \ T(p_i^a)||^2 \tag{1}$$

It is important that the class of deformations is chosen to match the expected deformations expected. If it is chosen too large it may easily generate false positives by deforming images outside the class of interest. The class of transformations, $\mathbf{T}$, will obviously be problem dependent. In our case we want to identify a specific pose for a certain person at different time instants. The transformations should then reflect the projected image shape variation between the different instances. This transformation will obviously be quite complex, involving motions of several body parts. As a simple preliminary measure we tried various linear transformations: pure translation, similarity and affine transformations. Pure translation gave the best results in terms of discriminating image frames. Therefore the transformation used is simply:

$$T(p_i^a) \ = \ p_i^a \ + \ t$$

and the matching distance was computed as the residual:

$$\min_{t} \frac{1}{n} \sum_{i=1}^{n} ||p_i^b - p_i^a - t||^2 \tag{2}$$

Evaluation of the complete algorithm of action recognition was made on a 30 sec sequence of tennis. The sequence contains forehand and backhand shots and mainly the player in a stand-by position.

The player was coarsely tracked automatically and a region of interest for each frame specified. In the upper half of this region Canny edge detection was applied. The edges were traced and subsampled to every fourth pixel. At each sample, the tangent direction was estimated. The number of edge points varied, in general, between $100 - 200$. No effort was made to delete the sometimes substantial number of background edges which occur in some frames. The upper half of the window is chosen as the upper body follows more consistent and distinctive patterns than the legs during different tennis strokes.

A specific frame (251 of the sequence) was selected as the key frame and the matching algorithm was applied to all frames in the sequence and a matching score was computed, see figure 4. There are clearly 9 local minima in the distance scores and each of these corresponds to the start of a forehand stroke as displayed.

4 Tracking by Point Transfer from Representative Key Frames

Motion capture requires body locations to be tracked over time, generally in multiple views in order to compute 3D representations of the body motion. In commercial and medical contexts this requires fitting visible markers to the person's body and recording with multiple calibrated cameras. Some attempts have been made to automate the tracking of body locations, notably rotational joints [16, 12], but no general solution has so far been presented. The problem is, of course, very difficult. This is partly because the points of interest are skeletal and are embedded within the body and a surface protrusion is all that is seen. This surface projection is therefore view dependent. Nevertheless, we believe that a crude estimation of these point can be tracked automatically. If this is done in multiple views, the 3D reconstruction problem will be overconstrained, allowing for filtering of tracking errors to some degree.

Without specific body markers, the projected image point to be tracked, for example the knee, may not have very much special *local* information to allow its determination. However, if correspondence for edge elements of the projected body has been computed between the image frame to be tracked and some frame in which the body locations are known a priori, this can be used to compute the body locations in the actual frame. This transfer of body locations could, of course, be from the previous frame in which they have been previously determined. This would imply, though, a standard open loop tracker where errors are propagated. Consider instead that a set of key frames of body postures are defined. In each of these frames the body locations are determined, most probably manually, at an earlier stage. Then if any frame in the actual sequence matches to at least one key frame, body locations can be transferred from the matched key frames. Thus a closed tracking loop has been established and tracking errors do not propagate indefinitely. Figure 7 illustrates how the correspondence field in figure 3 can be used selectively around specific body locations in order to transfer interior points between the two frames.

Tennis, as well as sports in general, is very repetitive. It consists of a limited repertoire of actions and often seen from a limited number of camera angles. Sports events are therefore an ideal environment for exploiting the key frame based mapping. In order for this to be possible, every frame in a sequence has be sufficiently close to some key frame. The action recognition presented in the previous section is a promising indication that this should be possible. We have demonstrated that it is possible to find the forehand shots in a sequence. This can be seen even more clearly from figure 5. The figure displays the results of calculating the similarity measure between a specific forehand (503-514) and all other forehand frames. For the illustrated matrix the darker the shade the lower the distance score. The repetitive nature of the forehand shot is clearly

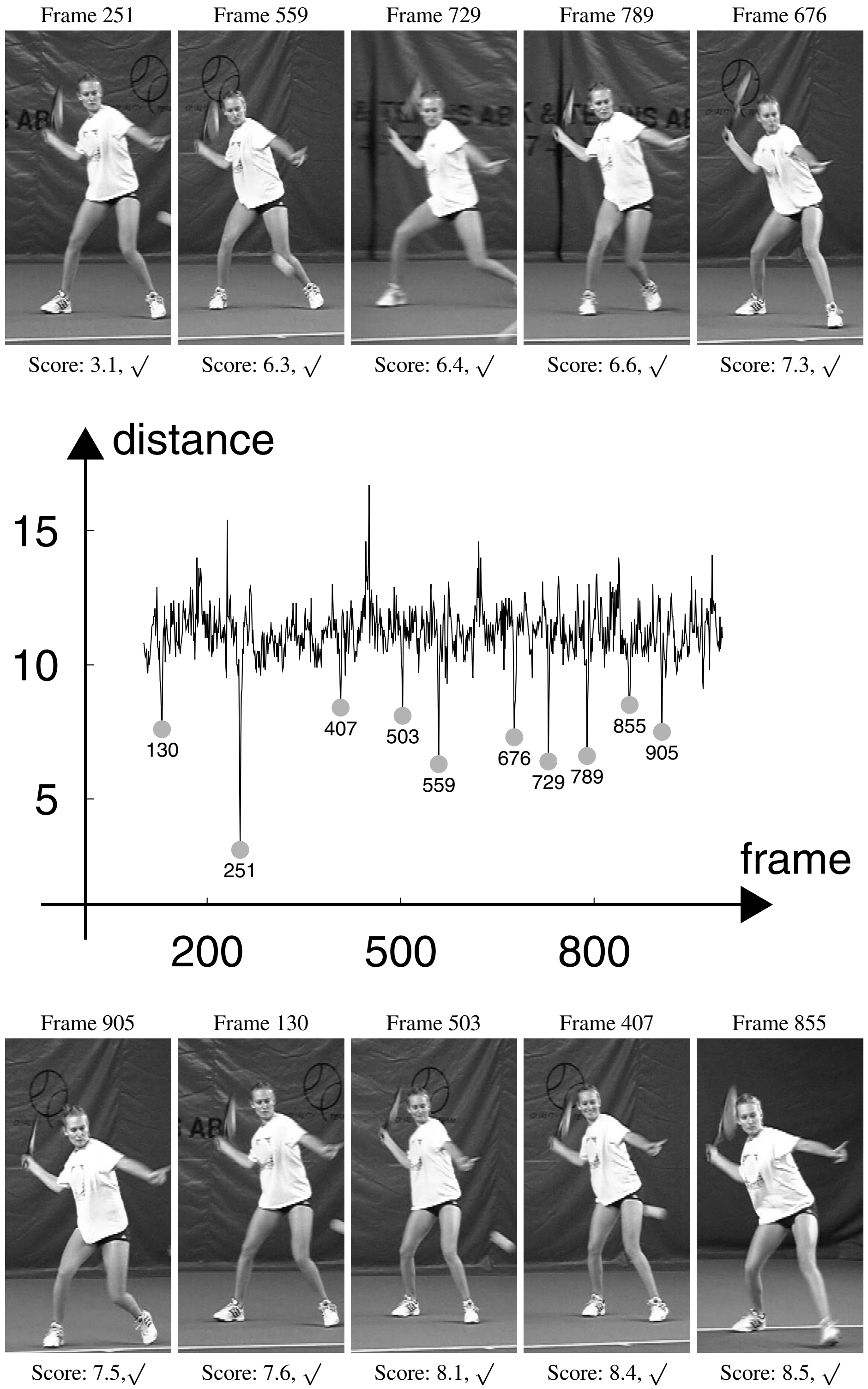


Fig. 4: Classified forehands for the sequence of the woman playing tennis. The displayed forehands correspond to the local minima of the distance scores.

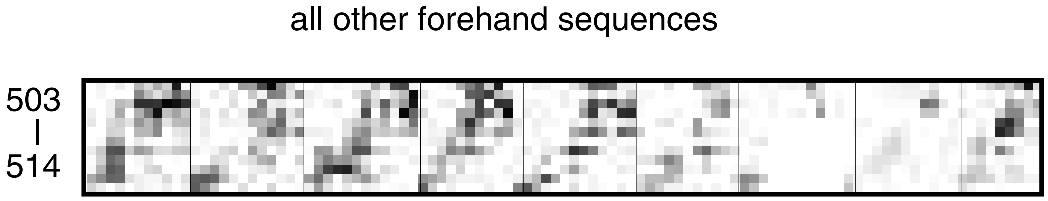


Fig. 5: Distance scores between forehand frames 503-514 and all other forehands.

seen in this matrix and it also indicates that finding a specific key frame for all frames in the sequence should be possible.

In figure 6 the distance scores for frames 503, 507, 510, and 513 to the frames in the forehand sequence 130 -140 are plotted. From this figure we can easily find the closest key frame for each frame. They appear in the correct order 503, 507, 510, and 513 as the frames 130-140 are traversed. In general an ordering of the key frames can be imposed which would simplify the key frame selection even further.

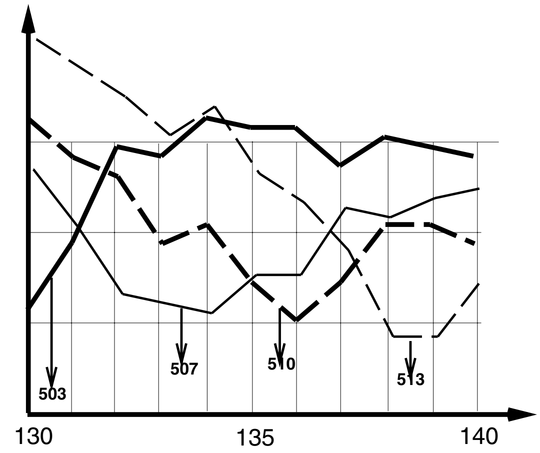


Fig. 6: Distance scores for frames 503, 507, 510, and 513 to all frames in forehand sequence 130 - 140.

5 Body Joint Localisation

In this section we focus on how the marked body locations can be transferred from the key frame to the image. The discussion can be split into two distinct parts. The first part describes how the body part locations are estimated just using the voting matrix V. In a completely constrained environment with the key frame being more or less exact replica of the image data this would probably be sufficient for successful localisation. However, as our data is not from such a world (though admittedly from a highly structured one) it is necessary to add another level of sophistication to the localisation. The other part is concerned with these issues. Priors are imposed upon the spatial arrangement of the points and also upon the anticipated colour of the intensity patches surrounding certain points. Crude tracking of the head and body regions is also implemented to give coarse prior estimates of their position. This latter information is used to refine the voting matrix V and to indirectly improve the estimates of the joint locations.

5.1 Point Transferral Using the Voting Matrix

Let $p^t_1, \ldots, p^t_n$ and $p^R_1, \ldots, p^R_n$ be the coordinates of the corresponding edge points in the test image, I_t, and the appropriately matched key frame image respectively. In the keyframe R the coordinates of the skeleton parts, P^R_k, are marked manually (see figure 10). For the upper-body keyframe these locations correspond to the left and right hand, elbow, and shoulders as well as the neck and the nose. These points are denoted by $\{P^R_k\}^K_{k=1}$. Let $P_k(t)$ be the location of the point corresponding to P^R_k in image I_t. It is assumed that a simple local translation, $\mathbf{v}_k(t)$, is sufficient to describe the transformation of the points from the key frame to the image:

$$P_k(t) = P^R_k + \mathbf{v}_k(t) \tag{3}$$

For each P^R_k a subset of the p^R_i's is chosen as support for this point from which $\mathbf{v}_k(t)$ is estimated, figure 7. This estimation is computed robustly to offset the influence of outliers.

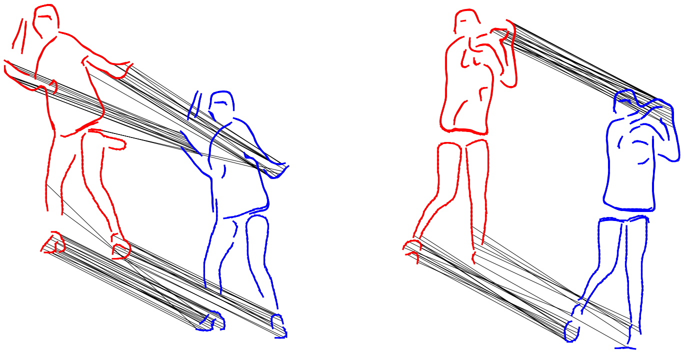


Fig. 7: Examples of selective correspondences used for point transfer from key frame to actual frame.

Coarse Head and Body Tracking: Updating the Voting Matrix The voting matrix V does not always give the correct correspondences. Thus if extra information can be obtained from other sources to adjust V that would be beneficial. An obvious candidate is to use the output of a conventional tracker.

Attempts to achieve automatic full body tracking on long sequences either using 3D fully articulate models or 2D appearance based ones have not meet with great success. However, tracking with less ambitious goals, for instance head and body localisation using colour information, have had very presentable results [7]. It was therefore decided to aid the matching process by tracking the head and body regions of the person. Each region is modelled as a quadrangle and a standard particle filter implemented for the tracking [2]. The likelihood function is based upon a sum-of-squares distance measure between a colour template and the image data. Examples of the localisation achieved are shown in figure 8.

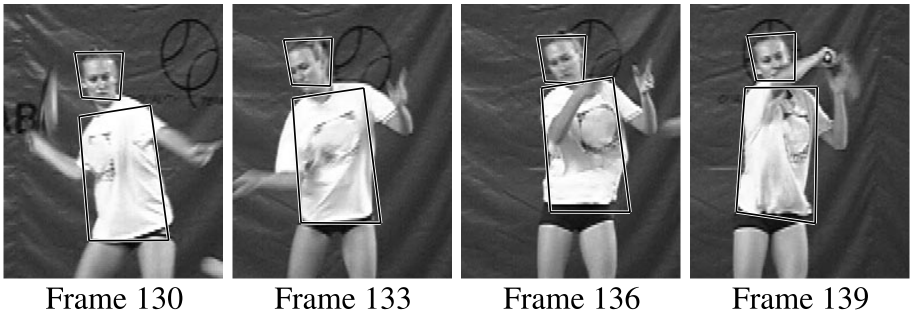


Fig. 8: Output of the coarse head and body tracker when applied to the tennis sequence. This output can be used to guide the correspondences computed. It is relatively easy to successfully to achieve this level tracking for long sequences.

With the successful completion of this part of the tracking the new information is incorporated into the voting matrix V. This is done as follows:

> Set $V(p_i^R, p_j^t) = 0$ if p_i^R and p_j^t are not close to the corresponding lines in corresponding matched quadrangles.

The above is not implemented for every point, but only for those points which are anticipated to have a consistent position with respect to the matched regions. For the upper-body example this corresponds to the points of the head and the shoulders and the torso. The points demarking the arms are omitted due to their varying relation to the body and head. The updated matrix can then be used to obtain estimates, $\{\mathbf{y}_k(t)\}$, of the interior points using the process previously described.

5.2 Incorporating Prior Constraints

The $\mathbf{y}_k(t)'s$ obtained will not in general be all inliers. This is mainly due to the fact that in the image data the edge information may be sparse in some areas. Numerous examples of this phenomenon can be seen in the pictures in figure 11 , especially with regard to the feet and the right arm. (Note the time component will be dropped from the subsequent notation as we are dealing soley with matching to individual images.)

Spatial Constraints A set of representative key frames has been chosen as being sufficient to represent the different topological shapes seen in a forehand stroke. Each key frame in the tracking process will be matched to an anticipated range of images. In these images the interior points should have a similar arrangement to the marked points in the key frame. This constraint can be mathematically stated using *Scaled Prismatic Models* of [14]:

$$P_{k+1} \sim f(P_{k+1}|P_k) = P_k + l_k(\cos(\theta_k)\sin(\theta_k))^T \tag{4}$$

for $k = 1, \ldots, K-1$ and where $l_k \sim N(l_k^R, \sigma_k^2)$, $\theta_k \sim N(\theta_k^R, \delta_k^2)$. P_0 follows the distribution $p_0(P)$. Equation (4) corresponds to a state evolution equation. The estimates $\{\mathbf{y}_k\}$ can be viewed as following the observation density:

$$\mathbf{y}_{k+1} \sim g(\mathbf{y}_{k+1}|P_{k+1}) = (1-\rho)N(P_{k+1}, \gamma^2 I_{2\times 2}) + \rho \mathrm{Un}(A) \tag{5}$$

where $0 \leq \rho \leq 1$ and $\mathrm{Un}(A)$ is a uniform distribution over a region A. From the chain model described and the measurements it is desirable to obtain estimates of the body joint locations from the expected value of the smoothing distribution:

$$\hat{P}_{1:K} = E\{p(P_{1:K}|\mathbf{y}_{1:K})\} \tag{6}$$

where $\mathbf{y}_{1:K} = (\mathbf{y}_1, \ldots, \mathbf{y}_K)$ and $P_{1:K} = (P_1, \ldots, P_K)$.

Finding the Smoothing Distribution Given the non-Gaussian nature of the state evolution equation and the observations, a Monte Carlo algorithm is chosen to produce random samples from the smoothing density of equation (6). A procedure as described in [10] is used. It is assumed that a weighted set of particles $\{P_k^{(i)}, \pi_k^{(i)}\}_{i=1}^N$ for $k = 1, \ldots, K$ which are drawn approximately from $p(P_k|\mathbf{y}_{1:k})$, have been obtained by applying a particle-filtering technique. Smoothing is then performed by backward simulation as follows:

- Start with the final particle set $\{P_K^{(i)}, \pi_K^{(i)}\}$ and choose a particle $\tilde{P}_K = P_K^{(i)}$ with probability $\pi_K^{(i)}$.
- The positions of the nodes of the chain are now chosen in reverse order. New weights are calculated
  - $\pi_{k|k+1}^{(i)} \propto \pi_k^{(i)} f(\tilde{P}_{k+1}|P_k^{(i)})$

  and selections made
  - Choose $\tilde{P}_k = P_k^{(i)}$ with probability $\pi_{k|k+1}^{(i)}$.

  The above is repeated for $k = K-1, \ldots, 1$ to obtain $(\tilde{P}_1, \tilde{P}_2, \ldots, \tilde{P}_K)$ an approximate realisation from $p(P_{1:K}|\mathbf{y}_{1:K})$.

This procedure results in choosing the particles that are simultaneously consistent with the state evolution equation and the measurements. When applied to enforce the spatial chain constraints isolated outliers are spotted and corrected accordingly, as shown by figure 9.

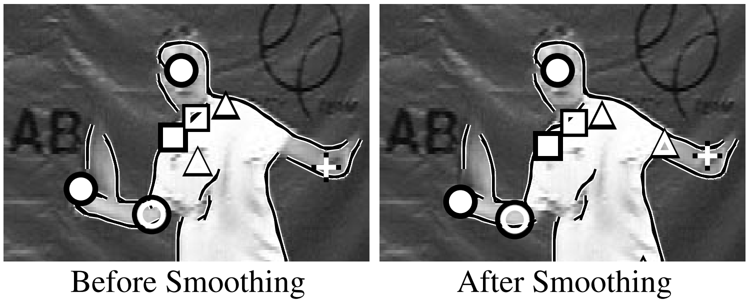


Fig. 9: Applying the spatial constraints allows the rectification of the position of the right elbow.

Colour Constraints In the keyframe each manually marked point P_k^R has an intensity patch associated with it. For example in figure 11 the left foot is surrounded by a predominantly white patch. This information can also be used to constrain the potential positions for the body joint locations and help to compensate for the cases when the edge data is insufficient and/or when the point matching partially fails. We exploit the information as follows.

An $L \times L$ template patch, $\mathbf{C}_k^R$, of RGB data centered at P_k^R is extracted from the key frame. For each particle $P_k^{(i)}$ a sum-of-squares distance score is made between the template and a patch from I_t centered at $P_k^{(i)}$ to obtain a measurement $z_k^{(i)}$. This measurement is turned into a probability:

$$p(z_k^{(i)}|P_k^{(i)}) = \exp -\lambda_0 z_k^{(i)} \tag{7}$$

The observation density equation (5) has to be updated to take account of this new measurement:

$$g'(\mathbf{y}_k, z_k^{(i)}|P_k^{(i)}) = g(\mathbf{y}_k|P_k^{(i)})p(z_k^{(i)}|P_k^{(i)}) \tag{8}$$

Particle filtering now results in each of the particle sets approximating $p(P_k|\mathbf{y}_{1:k}, z_{1:k})$ and the smoothing algorithm with samples drawn from the appropriately amended smoothing density.

Particle Filtering + Importance Function A few words must be made with regard to the implementation of the particle filter and obtaining the initial particle sets. It is well known that the use of importance sampling greatly increases the efficiency of the filtering process. The first problem is to find an initial particle set $\{P_0^{(i)}, \pi_0^{(i)}\}$ that is an accurate representation of $p(P_0|\mathbf{y}_0, z_0)$ given that $p(P_0)$ is typically a broad uniform distribution. Particles are placed in the vicinity of $\mathbf{y}_0$. However, it is not guaranteed that $\mathbf{y}_0$ is an inlier, therefore the search must be wider. This is achieved initially by considering each $\mathbf{y}_k$ as an inlier and drawing samples from $p(\mathbf{y}_0|P_k = \mathbf{y}_k)$, which can be calculated from state evolution equation. Also in each keyframe the line joining P_k^R to P_{k-1}^R defines an intensity profile. In the image if each $\mathbf{y}_k$ is considered as an inlier then a search can easily be performed to find the $\mathbf{y}_{k-1}$ that allows the best replication of the corresponding keyframe intensity profile. More particles are placed in this area.

The weights $\pi_0^{(i)}$ are updated appropriately to take into account the bias introduced. At each subsequent step the importance functions are normal distributions centred at $\mathbf{y}_k$ and the output of the profile intensity matching. ρ_1 of the particles are propagated using the state evolution equation while $(1-\rho_1)$ are chosen using the importance functions where $0 \leq \rho_1 \leq 1$.

6 Results

The techniques described were implemented on a forehand stroke that had been segmented out from the tennis sequence using the methods in section 3. The frames 130–140 were used as the test data. Hand-drawn key frames were constructed by tracing out the silhouettes of the tennis player from the frames 503, 506, 509 and 512. The resulting key-frames for the upper-body are shown in figure 10. These key frames encompass the range of topological types seen in a forehand stroke. However, the frame rate of the data is only 25Hz and the motion of the arm can be quite rapid in the middle part of the stroke. Thus for a different forehand we are quite likely to see images that are inbetween the key frames as the camera is synchronised differently with the test stroke to the key frame stroke. However, the matching process combined with the colour information partially compensates for the differences between the key frame and the data. The upper and lower body are treated separately, with the matching and the point transferral being computed independently. The key frame with the lowest distance score to the image frame was used to implement the point transferral. This search was also guided by a simple Markov chain to reduce the computational aspect and also to prevent inappropriate key frames being chosen.

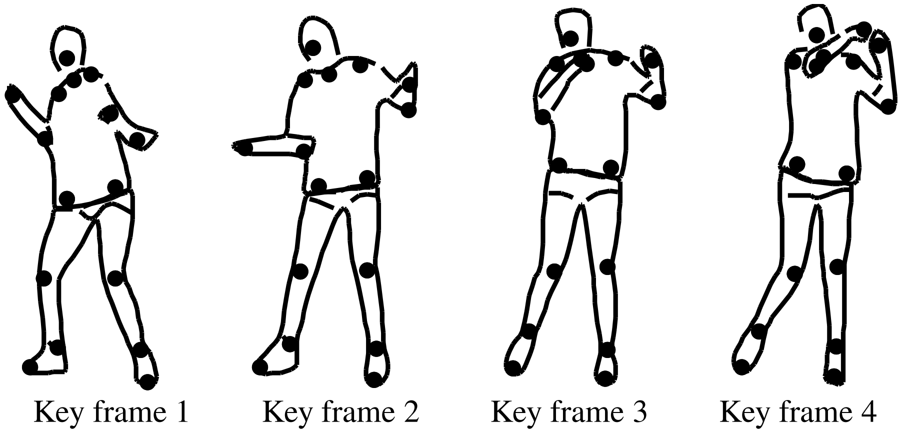


Fig. 10: Hand drawn key frames plus the manually marked interior points. These are the key frames used for matching to the forehand sequence.

Finally a note must be made about about how the parameters of the state evolution equation and the observation were set. As of yet no coherent strategy has been developed to answer the questions how many key frames are needed and to which range of poses can a key frame be matched. Thus the parameters were set by hand upon the anticipated answer to the latter question and were set to allow a quite a great deal of variation. Ideally these should be learned from training data. However, this awaits the decision whether it is better to add a key frame to the database of key frames or to improve the matching process. Inspection of the results obtained in figure 11 show very promising results on a very challenging sequence.

7 Conclusions

This paper presents a shift in the traditional approach to the tracking process. It is motivated by the desire to create a closed loop system. This is achieved by being able to recognise specific poses which correspond to stored key frames in which body joint locations have been previously defined. The matching process based upon the concept of *topological type* is the machinery that allows this approach to work. To improve the robustness of the point transferral, the spatial relationship of the body points and colour information is exploited. Experimental evidence has been presented which displays the power of our approach to segment out specific actions of interest, ie a forehand stroke and then to localise the body points in these highlighted frames. Therefore the need for explicit initialisation of a complicated model has been by-passed. Errors obtained in one frame do not propagate to the next frame. Obviously a number of issues remain unanswered and should be subject to further research. How many key frames are needed for each stroke ? Can generic key frames be defined that can be matched to different tennis players ? Can the parameters for the smoothing process be learnt systematically ? What range of actions can be successfully captured by this process ? We have so far demonstrated that four key frames can be used to map most of the body locations in a sequence of a forehand stroke to their correct positions. The ultimate number of keyframes necessary to map correct body locations in a whole tennis game remains to be determined. The option of selecting keyframes as a collection of body parts will also be investigated. In this way multiple keyframes can be generated by combinations of a limited set of parts.

Acknowledgements

This work was supported by the European project VIBES IST-2000-26001 and the Swedish Research Council.

References

1. S. Belongie and J. Malik. Matching with shape contexts. In *IEEE Workshop on Content-based Access of Image and Video Libraries*, June 2000.
2. A. Blake and M. Isard. *Active Contours*. Springer, 1998.
3. M. Brand. Shadow puppetry. In *Proc. 7th Int. Conf. on Computer Vision*, pages 1237–1244, 1999.

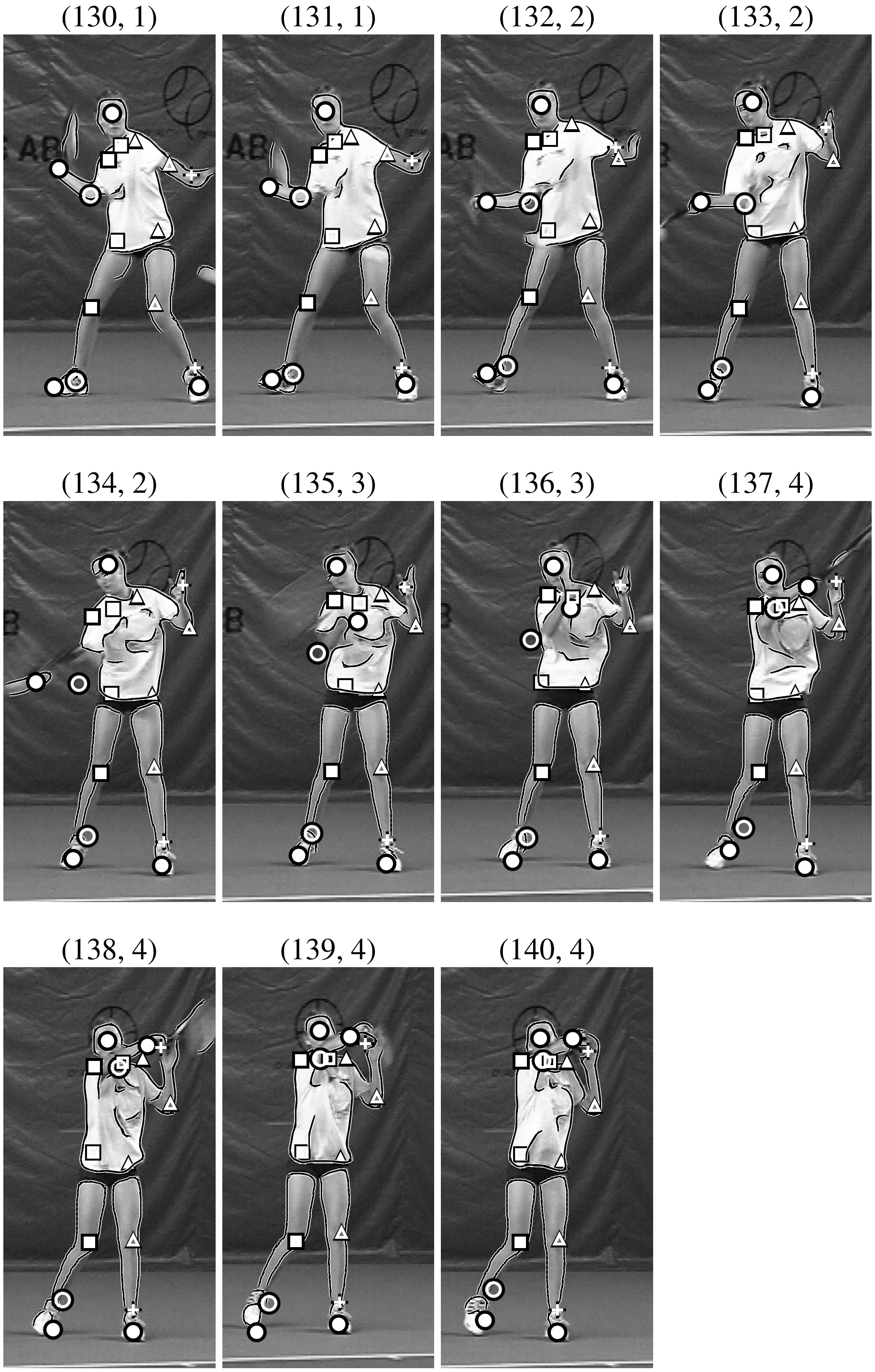


Fig. 11: Final estimates of the joint locations of the skeleton. Each point is displayed as a distinct shape. The edge data used in the matching process are also displayed. The numbers displayed correspond to the frame number of the sequence and to the matched key frame displayed in figure 10.

4. C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 8–15, 1998.
5. S. Carlsson. Order structure, correspondence and shape based categories. In *Shape Contour and Grouping in Computer Vision*, pages 58–71. Springer LNCS 1681, 1999.
6. S. Carlsson and J. Sullivan. Action recognition by shape matching to key frames. Workshop on Models versus Exemplars in Computer Vision at CVPR, 2001. Available at http://www.nada.kth.se/∼stefanc.
7. D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 142–149, Hilton Head Island, South Carolina, 2000.
8. J. Deutscher, A. Blake, and I. Reid. Motion capture by annealed particle filtering. *Proc. Conf. Computer Vision and Pattern Recognition*, 2000.
9. D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, January 1999.
10. S.J. Godsill, A. Doucet, and M. West. Methodology for monte carlo smoothing with application to time-varying autoregressions. In *Proc. International Symposium on Frontiers of Time Series Modelling*, 2000.
11. D. Hogg. Model-based vision: a program to see a walking person. *J. Image and Vision Computing*, 1(1):5–20, 1983.
12. N. R. Howe, M. E. Leventon, and W. T. Freeman. Bayesian reconstruction of 3d human motion from single-camera video. In S. A. Solla T. K. Leen and K-R. Muller, editors, *Advances in Neural Information Processing Systems 12*, 2000.
13. Y. Lamdan, J. Schwartz, and H. Wolfson. Object recognition by affine invariant matching. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 335–344, 1988.
14. D.D. Morris and J.M. Rehg. Singularity analysis for articulated object tracking. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 289–296, 1998.
15. N. Paragios and R. Deriche. Geodesic active regions for motion estimation and tracking. *Proc. 7th Int. Conf. on Computer Vision*, 1999.
16. J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. *Proc. 5th Int. Conf. on Computer Vision*, 1995.
17. K. Rohr. Towards model-based recognition of human movements in image sequences. *Computer Vision, Graphics and Image Processing*, 59(1):94–115, 1994.
18. H. Sidenbladh, M. Black, and D.J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *Poc of European Conference on Computer Vision*, pages 702–718, 2000.
19. K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *Proc. 8th Int. Conf. on Computer Vision*, July 2001.