

An Invariant Representation for Matching Trajectories across Uncalibrated Video Streams

Walter Nunziati¹, Stan Sclaroff², and Alberto Del Bimbo¹

¹ Dipartimento di Sistemi e Informatica - Università degli Studi di Firenze
{nunziati|delbimbo}@dsi.unifi.it

² Computer Science Department - Boston University
sclaroff@cs.bu.edu

Abstract. We introduce a view-point invariant representation of moving object trajectories that can be used in video database applications. It is assumed that trajectories lie on a surface that can be locally approximated with a plane. Raw trajectory data is first locally-approximated with a cubic spline via least squares fitting. For each sampled point of the obtained curve, a projective invariant feature is computed using a small number of points in its neighborhood. The resulting sequence of invariant features computed along the entire trajectory forms the view-invariant descriptor of the trajectory itself. Time parametrization has been exploited to compute cross ratios without ambiguity due to point ordering. Similarity between descriptors of different trajectories is measured with a distance that takes into account the statistical properties of the cross ratio, and its symmetry with respect to the point at infinity. In experiments, an overall correct classification rate of about 95% has been obtained on a dataset of 58 trajectories of players in soccer video, and an overall correct classification rate of about 80% has been obtained on matching partial segments of trajectories collected from two overlapping views of outdoor scenes with moving people and cars.

1 Introduction

Given a trajectory of a moving object acquired from a video sequence, we introduce a view-invariant representation of the trajectory based on algebraic projective invariants. Our envisioned use case is a video database application that returns all the objects whose trajectories are similar to a query trajectory, regardless of the view point from which the video has been taken. The user should be allowed to select both the object/trajectory of interest and the part of the trajectory to be used for the matching process. Examples of contexts that would benefit from such capabilities are sports videos and surveillance videos, where multiple cameras are usually deployed to cover the scene. Similarity could be measured across different views of the same object, for example to reconstruct the entire trajectory of the object throughout the scene, or across views of “similar” scenes, for example to retrieve players across multiple sports videos that move in similar way, allowing semantic event understanding.

More generally, this work is focused on analyzing multiple video streams captured from fixed cameras distributed in an indoor or outdoor environment, e.g., offices, classrooms, parking lots, a soccer field, etc. It is assumed that extrinsic/intrinsic calibration information for the cameras is not available, and it is not explicitly known if two or more cameras' fields of view actually overlap. Objects are assumed to move on surface that can be, at least locally, well approximated by a plane. Trajectories are acquired independently in each view, and for each trajectory its representation is based on projective invariant features measured at each observed point. For each point, the feature is computed using a small number of points in its neighborhood. The resulting sequence of invariant features computed along the entire trajectory forms the view-invariant descriptor of the trajectory itself. The time parametrization is exploited to compute (without ambiguity due to point ordering) the feature sequence. Once the descriptor is computed, it can be stored together with the trajectory it belongs to, to allow later retrieval. Since the descriptor is semi-local with respect to a point of the trajectory, partial matching can be performed using the relevant part of the descriptor. An example of this will be shown in Sect. 4.

To measure the similarity between two trajectory descriptors, a distance that takes into account the properties of the cross ratio is adopted. The proposed framework is tested both with synthetic data and with trajectories obtained from real videos, one from a surveillance dataset and the other from a soccer game. For each trajectory, we measure the distance with all the other trajectories for corresponding time segments. In quantitative evaluation, matching is performed with increasing levels of noise variance to verify the robustness of the method.

2 Related work

Several works have been proposed that investigate description, indexing and retrieval of video clips based on trajectory data. An important issue to be addressed is to provide a trajectory representation for which the effect of the perspective transformation due to the imaging process is minimized as much as possible. Earlier video database applications typically ignore this view-dependence problem, simply computing similarity directly from image trajectories [5, 4, 9]. More recent approaches have achieved some degree of invariance by using weak perspective models [2], or by recovering the image-to-world homography when an Euclidean model of the ground plane is available [14]. Such a method is not always viable, for example one could be interested to detect interesting patterns of people that move across public places, such as squares or stations, for which an Euclidean model of the scene could not be available. The proposed method allows to directly compare the projective invariant representation of each trajectory with either prototypes of interesting trajectories, for which their invariant representation has been precomputed, or with trajectories selected by the user.

In the context of video analysis for surveillance, trajectories have been used to align different views of the same scene using geometric constraints. In fact, it has been observed how trajectory data can be more reliable than static feature

points under wide variations in the viewpoint. In [10], objects are moving over a common ground plane which is captured from cameras with significant overlap, and the perspective plane correspondence is recovered using a robust estimation of homography between each camera pair. Here, moving objects are used as “markers” to recover point correspondences. Caspi and Irani [3] extended this approach to deal with non-planar trajectories, while also taking advantage of the temporal nature of the data. Their method recovers the fundamental matrix or the homography between two views, and can deal with asynchronous observations. Synchronized planar trajectories have been instead used in [15] to recover the correspondence model both for the cases of overlapping and non-overlapping cameras, to produce plausible homographies between two views. Each of the above methods explicitly recovers the geometric relation between different views, using either a homography or a fundamental matrix. Our method is suitable for solving a crucial step of all these approaches, which is to provide pairwise correspondence between trajectories, to initialize the registration algorithm. Furthermore, if the application only requires that each object is being stored with its (view-invariant) tracks, our representation can be used to this end without actually performing image registration.

Our approach is closely related to methods developed in the context of invariant model-based object recognition. Invariant theory is a classical mathematical theory, with results dating back to antiquity. Two invaluable references on the subject are [11, 12]. The method presented in [13], and recently used in [7], uses four points on a given object to establish a map with a canonical frame where a fifth point along the outline of the object has projective invariant coordinates. In [17], semi-differential invariants, constructed using both algebraic and differential invariants have been introduced. With respect to the above approaches, our method is more suited to the task of describing trajectories, in particular allowing for configurations of collinear points that often occur along trajectories.

3 View-invariant trajectory representation

We are given a set of time-indexed trajectories of the form $\mathbf{T} = \{\mathbf{p}(t_i)\}$, $\mathbf{p}(t_i) = (x(t_i), y(t_i))$, $i = [1 \dots n]$, where $(x(t_i), y(t_i))$ are image coordinates and $[t_i \dots t_n]$ are discrete time indices. It is assumed that at least locally, trajectories approximately lie on a planar surface. We want to derive a view point-invariant representation of such trajectories of the form $\xi(t_i)$, where each point is computed over a “small” neighborhood of $\mathbf{p}(t_i)$:

$$\xi(t_i) = f(\mathbf{p}(t_i - \delta t_i) \dots \mathbf{p}(t_i + \delta t_i)).$$

The function f must be invariant to planar projective transformations. Theoretically, given a curve in parametric form and its first eight derivatives, it is possible to find such signature in analytic form [18]. If the curve is given in implicit form, e.g. in the form $g(x, y) = 0$, at least four derivatives are necessary. Computing high order derivatives is known to be highly sensitive to noise. Since our data

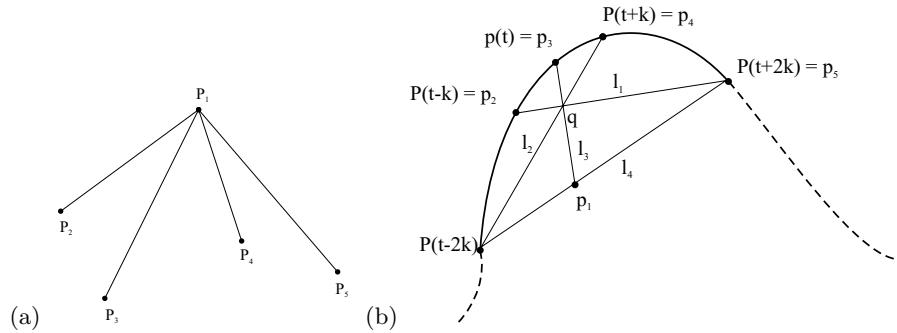


Fig. 1. a) 5 coplanar points that can be used to compute a cross ratio - b) The construction used in our method to compute cross ratios along the curve: $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$ are the points used to compute the cross ratio for $\mathbf{p}(t)$.

will come from a person or object tracking algorithm, we would need to fit high order parametric curves, which would be prone to over-fitting, especially in the case of simple, but noisy, trajectories. Given these considerations, we decide to use point-based projective invariants to avoid the problem of fitting high order curves.

The most fundamental point-based projective invariant in the plane is the cross ratio of five coplanar points, no three of which are collinear (Fig. 1(a), see also [11], Chapter 1). Two independent cross ratios can be computed from this configuration. If points are expressed in homogeneous coordinates, the cross ratio takes the form:

$$\tau = \frac{|m_{125}| |m_{134}|}{|m_{124}| |m_{135}|} \quad (1)$$

where $m_{ijk} = (\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k)$ with $\mathbf{p}_i = (x(t_i), y(t_i), 1)^t$ and $|m|$ is the determinant of m . The point \mathbf{p}_1 is the *reference point*. If points $\mathbf{p}_2 \dots \mathbf{p}_5$ are collinear, the cross ratio becomes independent of \mathbf{p}_1 , and it is reduced to the cross ratio of the distances between points on the segment joining \mathbf{p}_2 and \mathbf{p}_5 . Under planar perspective transformations, the cross ratio (1) is unchanged. However, its value depends on the order of the points used to compute it; for instance: $\tau(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5) \neq \tau(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_5, \mathbf{p}_4)$. This is a serious issue in model-based object recognition, since usually point correspondences are unknown, and one needs to rely on projective and permutation invariant features. Although such features have been derived [16], it is known that permutation invariant features turn out to be considerably less stable and less discriminative than features computed on labeled points.

Since in our case trajectories are time-indexed sets of points, we have a natural parametrization that allows us to compute the cross ratio using a predefined point ordering. However, choosing the points along the trajectory to be used in the cross ratio is non trivial, since we need to ensure that at least the reference point is not aligned with the other points, otherwise the cross ratio is undefined. A potential solution is to choose points $\mathbf{p}_2 \dots \mathbf{p}_5$ on the trajectory, and \mathbf{p}_1 off

the trajectory, such that even if $\mathbf{p}_2 \dots \mathbf{p}_5$ are aligned, the cross ratio can still be computed and reduces to the cross ratio of four collinear points under a suitable choice of the point order. However, to obtain a consistent feature, the point \mathbf{p}_1 must be chosen according to a projective invariant construction, otherwise a feature computed using an arbitrary point off the trajectory would be just meaningless.

A simple but effective method is sketched in Fig. 1(b) and detailed in Algorithm 1. For each point $\mathbf{p}(t_i)$ along the curve, four other points $\mathbf{p}(t_i - 2k)$, $\mathbf{p}(t_i - k)$, $\mathbf{p}(t_i + k)$, $\mathbf{p}(t_i + 2k)$ are used to compute the representation value of the current point. k is a time interval that controls the scale at which the representation is computed. The greater is k , the less local the representation. The points are first locally smoothed using a cubic spline fitted via least squares. If $(x_r(t_i), y_r(t_i))$ are the raw data, the local feature is computed with points of the form $(x_s(t_i), y_s(t_i))$ obtained from the fitted spline at corresponding time indices.

This construction can always be computed, provided that there are no four collinear points. With respect to of Fig. 1(a), if points $\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$ are collinear, then the cross ratio becomes independent of the choice of \mathbf{p}_1 . Hence, if collinearity is detected, we simply use the collinear points to compute a 4-point cross ratio. If collinearity is not detected, points $\mathbf{p}(t_i - 2k)$, $\mathbf{p}(t_i - k)$, $\mathbf{p}(t_i + k)$, $\mathbf{p}(t_i + 2k)$ are used to compute the point \mathbf{q} , and then the intersection between the lines defined by segments $\mathbf{p}(t_i), \mathbf{q}$ and $\mathbf{p}(t_i - 2k), \mathbf{p}(t_i + 2k)$ is chosen to be the reference point for the cross ratio. Being based on collinearity and intersection between points, the construction is obviously projective invariant. The projective invariant representation of the trajectory is the sequence of the cross ratios computed along the trajectory at each t_i . The parameter k controls the locality of the representation. In principle, a small k is desirable, since it would give a more local representation for matching partial trajectory segments. However, this must be traded-off with the informative content of the resulting transformed sequence, since on smaller scale the cross ratios tend to assume very similar values. In our experiment, we verified that for objects like people and cars, a good choice is to select k approximately equal to the observation rate.

3.1 Comparing trajectories

In [1], it is shown that a probability density function for the cross ratio can be computed in closed form, together with the corresponding cumulative density function. A distance measure derived from this function has been proposed in [8] in the context of object recognition. This measure has the property of stretching differences of cross ratios of big values, which are known to be less stable. Moreover, it takes into account the symmetric properties of cross ratios, in particular the fact that there are two ways to go from one cross ratio to another: one passing through the real line, and the other through the point at infinity. We have verified experimentally that the invariant feature described above obeys the distribution derived in [1], although input points are not exactly independent.

Algorithm 1 Computing the feature for a point $\mathbf{p}(t_i)$

$\mathbf{p}(t_i)$ the current point, obtained from the local spline approximation of the raw data ($i = [1 \dots n]$); k predefined time interval
 $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$ the points used for computing the cross ratio
 $\mathbf{p}_2 \leftarrow \mathbf{p}(t - k), \mathbf{p}_3 \leftarrow \mathbf{p}(t), \mathbf{p}_4 \leftarrow \mathbf{p}(t + k), \mathbf{p}_5 \leftarrow \mathbf{p}(t + 2k)$
if $\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$ are collinear **then**
 Compute the cross ratio of four collinear points using $\mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$
 $\xi(t_i) = \frac{|\mathbf{p}_2 - \mathbf{p}_5| |\mathbf{p}_3 - \mathbf{p}_4|}{|\mathbf{p}_2 - \mathbf{p}_4| |\mathbf{p}_3 - \mathbf{p}_5|}$
else
 $\mathbf{l}_1 = \mathbf{p}(t - k) \times \mathbf{p}_5$ line through $\mathbf{p}(t - 2k)$ and \mathbf{p}_5
 $\mathbf{l}_2 = \mathbf{p}(t - 2k) \times \mathbf{p}_4$ line through $\mathbf{p}(t - 2k)$ and \mathbf{p}_4
 $\mathbf{q} = \mathbf{l}_1 \times \mathbf{l}_2$ intersection between \mathbf{l}_1 and \mathbf{l}_2
 $\mathbf{l}_3 = \mathbf{p}(t_i) \times \mathbf{q}$ line through $\mathbf{p}(t_i)$ and \mathbf{q}
 $\mathbf{l}_4 = \mathbf{p}(t - 2k) \times \mathbf{p}_5$ line through $\mathbf{p}(t_i - 2k)$ and \mathbf{p}_5
 $\mathbf{p}_1 = \mathbf{l}_3 \times \mathbf{l}_4$
 Compute the cross ratio of $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$
 $\xi(t_i) = \frac{|m_{125}| |m_{134}|}{|m_{124}| |m_{135}|}$
end if

Hence, to compare two cross ratios τ_1 and τ_2 , we use their distance with respect to the cumulative distribution function:

$$d(\tau_1, \tau_2) = \min(|F(\tau_1) - F(\tau_2)|, 1 - |F(\tau_1) - F(\tau_2)|)$$

where $F(x)$ is defined as follows:

$$F(x) = \begin{cases} F_1(x) + F_3(x) & \text{if } x < 0 \\ 1/3 & \text{if } x = 0 \\ 1/2 + F_2(x) + F_3(x) & \text{if } 0 < x < 1 \\ 2/3 & \text{if } x = 1 \\ 1 + F_1(x) + F_2(x) & \text{if } x > 1 \end{cases}$$

$$F_1(x) = \frac{1}{3} \left(x(1-x) \ln\left(\frac{x-1}{x}\right) - x + \frac{1}{2} \right),$$
$$F_2(x) = \frac{1}{3} \left(\frac{x - \ln(x) - 1}{(x-1)^2} \right),$$
$$F_3(x) = \frac{1}{3} \left(\frac{(1-x) \ln(1-x) + x}{x^2} \right).$$

Given two trajectories $\mathbf{T}_1 = (x_1(t_i), y_1(t_i))$, $\mathbf{T}_2 = (x_2(t_i), y_2(t_i))$ and the corresponding invariant representation $\xi_1(t_i)$, $\xi_2(t_i)$, their distance is defined as follows:

$$D(\mathbf{T}_1, \mathbf{T}_2) = \sum_{i=1}^n d(\xi_1(t_i), \xi_2(t_i)).$$

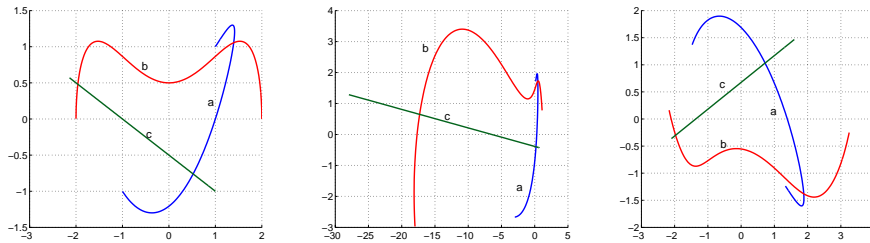


Fig. 2. Three views of three of the synthetic trajectories used to test the algorithm.

4 Experimental results

The proposed method has been tested on three different sets of data. In the first experiment, we generated several planar trajectories, and we applied two different homographies to obtain the views shown in Fig.2 for three sample trajectories. Each trajectory was uniformly sampled in the first view, and then the “observed” points were projected into the other views and corrupted with Gaussian noise to simulate the effect of the measurement error. Each curve consisted of about 300 points, and we set $k = 10$. This value was appropriate to capture the overall shape of the trajectory in the neighborhood of a given point.

The experiment was repeated for increasing levels of the noise variance, up to approximately 20% of the average distance between points. Up to this level, it was observed that the method is always able to recover the correct match, while further increasing the amount of noise produced correspondences that were no longer valid.

In the second experiment, we used a dataset made available for the VS-PETS 2001 workshop¹. It consists of a video from a soccer game, taken from a fixed position. There are 58 trajectories in this dataset, although some of them are very short and have not been considered for the matching test. Two views of the trajectories were generated from the data, and noise was added independently to simulate the effect of measurement error. For this and the following experiment, we set $k = 25$. We verified experimentally that this value is suitable for trajectories that shown a sufficient degree of variability for our method, such as those of players in a soccer game. Fig.3 shows the results obtained for different level of noise, up to 10% of the average distance between points on the trajectories. The correct overall classification rate, ($correct/total$) was 95%, 81% and 65% respectively. As can be expected, it was observed that the more long and varying the trajectory is, the more robust the match.

In the third experiment, we used another dataset from the VS-PETS workshop. In this dataset, two cameras observe the same outdoor scene from two widely separated points of view with a significant overlap (Fig.4). The scene features a number of moving persons and cars. Time-aligned positions of the

¹ <http://peipa.essex.ac.uk/ipa/pix/pets/PETS2001/DATASET1/>

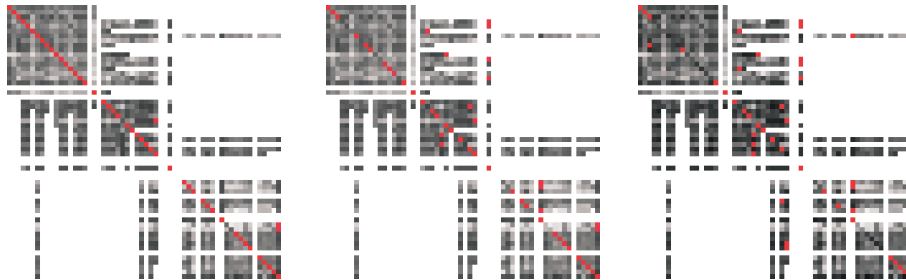


Fig. 3. From left to right: results obtained from the soccer dataset for Gaussian noise with variance 0, 5% and 10% of the average distance between points. Element i, j of the matrix is the distance between trajectories i and j (darker means closer). The red square on each line indicates the best match. White lines correspond to very short trajectories that have not been used for matching.

image-centroid are provided for each object through the entire sequence for both views. This was the most challenging experiment because most trajectories take place in the region of overlap only for a short time. We used the same approach described in the previous experiments to recover similarity between trajectories across views, except that this time artificial noise was not added since with independent tracking data in both views was provided. Moreover, trajectories were compared only using the part of the descriptor related to their common temporal support, to verify the performance in the case of partial matching. The results are shown in Fig. 5(a) in the form of a distance matrix, where intensity level are distance measures (darker means closer). It can be seen that the correct correspondence was almost always the best match; the overall correct classification rate was about 80%. It is also interesting to notice how similar trajectories can be clearly identified in the distance matrix with a connected block of low-distance values; for example, trajectories 3, 4 and 5 come from observing people walking together, and so do trajectories 8 and 9.

In two cases the matching method failed (trajectories 13 and 14, highlighted with crosses in Fig.5). The first false match was due to a trajectory of a person suddenly turning and walking back. This introduced a discontinuity in the trajectory that was not reflected in the corresponding invariant representation. In the second case, the object appeared in the region of overlap for a very limited time, hence the observed trajectory was too short to be distinctive.

5 Discussion

We proposed an algorithm that matches trajectories of objects moving over a locally-planar surface across different perspective views. We derived a trajectory representation based on projective invariant features that can be computed using information extracted only from the trajectory itself. The distance measure from

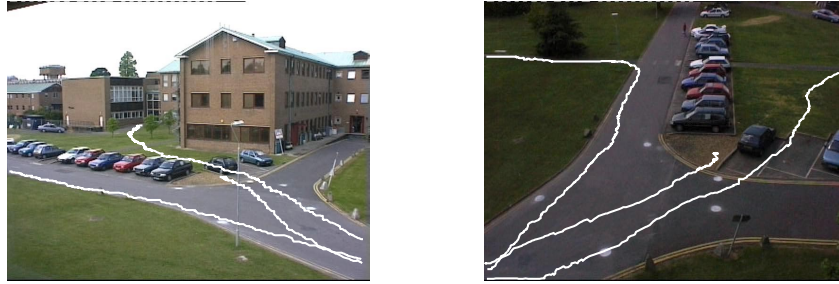


Fig. 4. Examples of correctly matched trajectories from the surveillance videos superimposed on the background image of the two views.

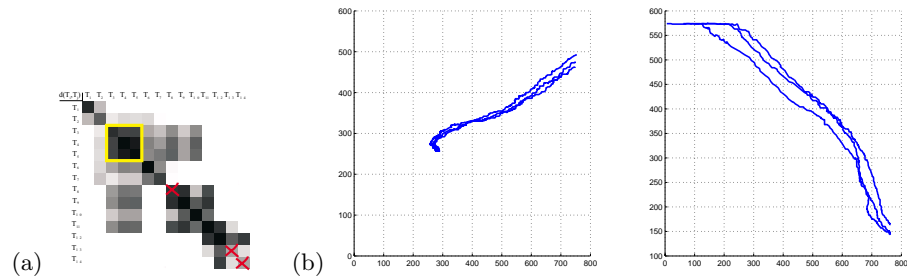


Fig. 5. a) Distances between time-aligned trajectories across the two views. Darker means closer, crosses indicate failed cases - b) Trajectories corresponding to the group highlighted in yellow in the distance matrix.

two trajectories is derived from the distance between two cross ratios, which in turn is related to the probability density function of the cross ratio.

Preliminary experimental results showed that the algorithm is quite robust to noise in the case of synthetic generated data, and that it can reliably discover similarity between real world trajectories, such as those of people or cars.

Since the algorithm is based only on information extracted from the trajectory, a potential problem may arise in scenes where multiple objects move on similar trajectories at similar speed (for instance, pedestrians walking across a square). In this situation, the algorithm cannot differentiate between trajectories. To overcome this problem, other features should be considered, in particular those based on object's appearance such as proposed in [6].

Several other improvements could be made to the basic algorithm. For example, matching trajectories across different but similar video streams would benefit from a similarity measure performed at different scales, whereas the current formulation operates at only one scale. At the coarser scale, the descriptor would capture the overall shape of the trajectory, ruling out obvious false matches, while decreasing the value of k would help to discriminate between trajec-

ries at a finer level. In the case of streams obtained from different views of the same scene, it would be interesting to recover the time alignment if this is not provided, in particular under conditions of partial overlap.

Acknowledgments

This work was supported in part by the U.S. Office of Naval Research, grant N00014-03-1-0108, and it was carried out while Walter Nunziati was visiting the Image and Video Computing Group at Boston University.

References

1. K. Åstrom and L. Morin. “Random Cross Ratios”. *Report RT 88 IMAG-LIFIA*, 1992.
2. F. Bashir, A. Khokhar, and D. Schonfeld. “A hybrid system for affine-invariant trajectory retrieval”, *Proc. of ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2004.
3. Y. Caspi, D. Simakov, and M. Irani. “Feature-Based Sequence-to-Sequence Matching”. *Proc. of VMODS Workshop*, 2002.
4. W. Chen, S.-F. Chang. “Motion Trajectory Matching of Video Objects”. *Proc. of Storage and Retrieval for Media Databases*, 2000.
5. S.-F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong. “VideoQ: An Automatic Content-Based Video Search System Using Visual Cues”. *Proc. of ACM Multimedia*, 1997.
6. A. Efros, A. Berg, G. Mori and J. Malik. “Recognizing Action at a Distance”. *Proc. of ICCV*, 2003.
7. R. Fergus, P. Perona, and A. Zisserman. “A Visual Category Filter for Google Images”. *Proc. of ECCV*, 2004.
8. P. Gros. “How to Use the Cross Ratio to Compute Projective Invariants from Two Images”. *Proc. of Application of Invariance in Computer Vision*, 1993.
9. V. Kobla, D. Doermann, and C. Faloutsos. “VideoTrails: representing and visualizing structure in video sequences”. *Proc. of ACM Multimedia*, 1997.
10. L. Lee, R. Romano, and G. Stein. “Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame”. *IEEE TPAMI*, 2000.
11. J. Mundy and A. Zisserman, editors. “Geometric Invariance in Computer Vision”. MIT Press, Cambridge, MA, 1992.
12. J. Mundy and A. Zisserman, editors. “Applications of Invariance in Computer Vision”. Springer LNCS, 1994.
13. C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy. “Planar Object Recognition Using Projective Shape Representation”. *International Journal of Computer Vision*, 1995.
14. Shim, C.B., Chang, J.W. “Efficient similar trajectory-based retrieval for moving objects in video databases”. *Proc. of CIVR*, Springer LNCS, 2003.
15. Chris Stauffer, Kinh Tieu. “Automated multi-camera planar tracking correspondence modeling”. *Proc. of CVPR*, 2003.
16. T. Suk and J. Flusser. “Point projective and permutation invariants”. *Proc. of Computer Analysis of Images and Patterns*, Springer LNCS, 1997.
17. L. Van Gool, P. Kempenaers, and A. Oosterlinck. “Recognition and semi-differential invariants”. *Proc. of CVPR*, 1991.
18. Isaac Weiss. “Differential invariants without derivatives”. *Proc. of IEEE ICIP*, 1992.