Tracking by Cluster Analysis of Feature Points using a Mixture Particle Filter

Wei Du Justus Piater
University of Liege, Department of Electrical Engineering and Computer Science,
Institut Montefiore, B28, Sart Tilman Campus, B-4000 Liege, Belgium
weidu@montefiore.ulg.ac.be, *justus.piater@ulg.ac.be*

Abstract

A moving target produces a coherent cluster of feature points in the image plane. This motivates our novel method of tracking multiple targets via feature points. First, the Harris corner detector and the Lucas-Kanade tracker are applied in each frame to detect feature points and their associated velocities. Points that are both spatially co-located and exhibit similar motion are grouped into clusters. Due to the non-Gaussian distribution of the points in a cluster and the multi-modality resulting from multiple targets, a special particle filter, the mixture particle filter, is adopted to model the mixture point distribution over time. Each cluster is treated as a mixture component and is modeled by an individual particle filter. The filters in the mixture are instantiated and initialized by applying the EM algorithm, are reclustered by merging overlapping clusters and splitting spatially disjoint clusters, and are terminated when their component weights drop below a threshold. The advantage of using mixture particle filtering is that it is capable of tracking multple targets simultaneously and also of handling appearing and disappearing targets. We demonstrate the effectiveness of our method on different PETS datasets.

1. Introduction

Feature point tracking is essential to many computer vision tasks such as image mosaicking [1], structure from motion [2], object tracking [3], etc. This paper presents a novel method of tracking multiple targets via feature points. The principle behind it is simple. As a moving target produces a coherent cluster of feature points in the image plane, tracking is converted to cluster analysis by perceptually grouping those groups of feature points that are spatially co-located and exhibit similar motion as well. Throughout the paper, the term *cluster* denotes a set of coherent points coming from a moving target or a group of moving targets.

An intuitive idea of clustering feature points is to construct finite Bayesian finite mixture models. By assuming Gaussian distribution, it can be easily solved using the EM algorithm [4] . However, the Gaussian assumption does not hold in practice. In fact, the spatial distribution of the points in a cluster is better approximated by a finite uniform model. Therefore, a particle filter, which represents the posterior distribution with a set of weighted particles and makes no assumption at all about the model of the distribution, is suitable for handling non-Gaussianity [5]. Traditional particle filters perform poorly at consistently maintaining the multi-modality in the target distribution that often results from multiple targets. Vermaak *et al.* [6] introduced a mixture particle filter (MPF), where each component (mode or, in our case, cluster) is modeled with an individual particle filter that forms part of the mixture. The filters in the mixture interact only through the computation of the component weights. By delegating the resampling step to individual filters, one avoids the problem of sample depletion, which is often responsible for the loss of a track [7].

In this paper, we extend the MPF filtering approach to fit our cluster analysis task. First, feature points and their associated velocities are detected in each frame using the Harris corner detector [8] and the Kanade-Lucas-Tomasi tracker (KLT) [9]. The MPF is then applied to model the mixture point distribution over time. Each cluster is treated as a mixture component and is tracked by an individual particle filter. The filters in the mixture are instantiated and initialized by applying the EM algorithm, are reclustered by merging overlapping clusters and splitting spatially disjoint clusters, and are terminated when their component weights drop below a threshold. The advantage of using MPF is that it is capable of tracking multiple targets simultaneously and of handling appearing and disappearing targets. Our method is especially well suited for the typical video surveillance configuration where the cameras are still and targets of interest appear relatively small in the image so that feature points on them show strong coherence in space and motion. We demonstrate the effectiveness of our method on different PETS datasets [10].

Most previous work on point tracking focused on reconstructing individual point trajectories as long as possible. For instance, the KLT algorithm matches points by minimizing the sum of squared intensity differences [9]. As minimization is sensitive to local extrema, KLT fails easily in

This work has been sponsored by the Région Wallonne under DGTRE/WIST contract 031/5439.

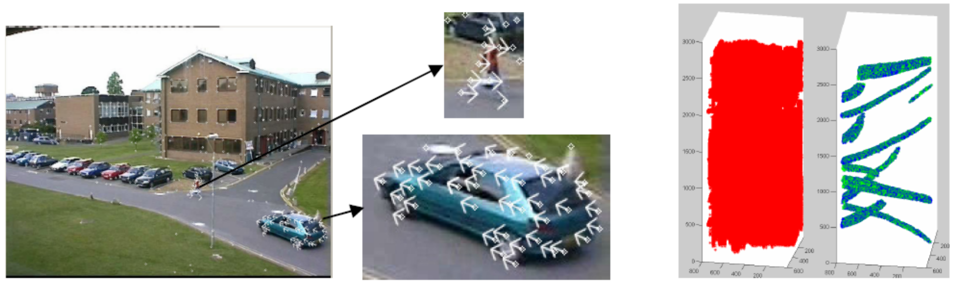

Figure 1: Result of Harris corner detection and KLT tracking. In the left panel, point distributions of clusters are shown in the image plane. In the right panel, all the corners in the sequence are displayed in the spatio-temporal domain. After removing background points, the structure of the trajectories of moving targets are clearly seen.

case of occlusions and target deformations. Robust methods such as optimal matching [11], probabilistic filtering techniques [12], model-based approaches [13], etc. have been developed to improve the reliability of single-point tracking. However, the key problem remains: When occlusions or deformations occur, feature points become less stable – corner points are occluded or may turn into edges – making tracking or matching difficult. With our approach, missing or unstable feature points will not affect the tracking results very much as we consider only stochastic properties of the clusters of feature points. A similar idea was introduced by Pece [14] where tracking was done by cluster analysis of regions using the EM algorithm. Borrowing only the idea of cluster analysis, our contributions are, first, to apply it to points instead of regions, thus avoiding background modeling which is sensitive to illumination changes; second, to take motion coherence into account when computing measurements of clusters, which improves the robustness of cluster analysis; third, to integrate the prior motion model and measurements from the image using the MPF, which significantly stabilizes the estimation of the cluster parameters.

The rest of the paper is organized as follows. Section 2 describes the MPF model and our point cluster model. Automatic initialization by EM based cluster analysis is given in Section 3. Section 4 introduces the mixture particle filtering process. Results on sequences from PETS2001 are illustrated in Section 5.

2. Model Description

The motivation of this work is to develop a multi-target tracker based on point tracking for video surveillance applications. By detecting Harris corners and applying KLT in each frame, a number of feature points with their associated velocities in the sequence are obtained, as shown in Figure 1. Points on moving targets exhibit large displacements, whereas points on the static background are characterized by very little motion. Our task is to detect how many clusters are present, and to assign each point to a cluster.

2.1. MPF Model

As stated above, MPF is adopted to solve the non-Gaussianity and multi-modality. In state-space models, the state sequence $\{x_t\}$ is assumed to be a hidden Markov process with a prior dynamic model $D(x_t|x_{t-1})$. The observations up to time t $y^t = \{y_1, \ldots, y_t\}$ are conditionally independent given the process $\{x_t\}$ with marginal distribution $p(y_t|x_t)$. For tracking, the distribution of targets of interest is the filtering distribution $p(x_t|y^t)$, which can be computed recursively:

$$\textbf{Prediction}: p(x_t|y^{t-1}) = \int D(x_t|x_{t-1})p(x_{t-1}|y^{t-1})\mathrm{d}x_{t-1}$$

$$\textbf{Update}: \quad p(x_t|y^t) = \frac{p(y_t|x_t)p(x_t|y^{t-1})}{\int p(y_t|s_t)p(s_t|y^{t-1})\mathrm{d}s_t}$$

To capture multi-modality, the filtering distribution in MPF is reformulated as an M-component mixture model

$$p(x_t|y^t) = \sum_{m=1}^{M} \pi_{m,t} p_m(x_t|y^t)$$

where $\pi_{m,t}$ is the weight of the m-th component and $\sum_{m=1}^{M} \pi_{m,t} = 1$. Assuming that the filtering distribution at the previous step $p(x_{t-1}|y^{t-1})$ is known, the new prediction and filtering distributions are obtained as

$$\textbf{Prediction}: p(x_t|y^{t-1}) = \sum_{m=1}^{M} \pi_{m,t-1} p_m(x_t|y^{t-1})$$

where $p_m(x_t|y^{t-1})$ is the prediction distribution for the m-th component, and

$$\textbf{Update}: \quad p(x_t|y^t) = \sum_{m=1}^{M} \pi_{m,t} p_m(x_t|y^t)$$

where $p_m(x_t|y^t)$ is the filtering distribution for the m-th component, and the new component weight $\pi_{m,t}$ is computed from its previous weight and the particle weights.

This is an elegant result and means that the filtering recursion can be performed for each component individually. Hence in MPF, each component is modeled with an individual particle filter. The filters in the mixture interact only through the computation of the component weights. By distributing the resampling step to individual filters, the multi-modal distribution is maintained during the propagation in time. Consult Vermaak *et al.* [6] for more detail.

2.2. Point-Cluster Model

We modify the MPF model to fit our specific problem. The observations are feature points detected in the sequence. A feature point x_i is defined by its image coordinates u_i and its velocity s_i. Clustering feature points amounts to modeling the mixture point distribution. Assuming we have the initial mixture distribution, feature points can be associated with one of the M clusters. Let $X_t^m = \{x_{i,t}^m, i = 1, \ldots, n_m\}$ be the feature points in cluster m at time t.

To apply MPF, a particle filter is instantiated for a cluster when it is detected. A set of particles are sampled from the cluster in position-velocity space. Let $X_t'^m = \{x_{i,t}'^m, i = 1, \ldots, n'_m\}$ be the particles in filter m corresponding to cluster m at time t. The particles are then propagated, measured, and resampled in the classical manner. Distinct filters only interact when updating their component weights. To handle occlusions and appearing and disappearing targets, merges and splits are performed when clusters overlap or become too dispersed, respectively, as is done by Vermaak *et al.* [6]. In this way, the mixture point distribution, represented by weighted particle sets, is propagated in time. Let $C_t^m = \{X_t^m, X_t'^m, W_t^m, \pi_t^m\}$ be the cluster representation, where $W_t^m = \{w_{i,t}^m, i = 1, \ldots, n'_m\}$ is the set of particle weights, and π_t^m is the component weight of the cluster.

Note that it is the moving targets, represented by the clusters, that we are tracking instead of individual feature points. Therefore, the real state of a target in the filter is the distribution of the particles in the cluster, parameterized by a Gaussian, $\lambda_t^m = \{o_{m,t}, \Sigma_{m,t}^o, v_{m,t}, \Sigma_{m,t}^v\}$, where $o_{m,t}$ is the spatial center, $\Sigma_{m,t}^o$ is the spatial covariance, $v_{m,t}$ is the average velocity, and $\Sigma_{m,t}^v$ is the velocity covariance. The position and velocity distributions are assumed to be independent.

3. EM based Cluster Analysis

Automatic initialization is crucial to the success of a video surveillance system. Targets should be detected and located when they first appear. An EM algorithm is applied when a large number of feature points exist that are not associated with any existing clusters. Note that new targets may not only occur at the borders but anywhere within the image.

Deciding the number of clusters in the data is usually the hardest problem in clusters analysis. A voting technique was devised to solve this problem. Intuitively, each point spreads a weight to its neighbors based on the distance between them. After voting, each point computes its weight by collecting all the votes received. Points near the center of a cluster tend to have a larger weight. This method is incidentally the first phase ("sparse voting") of tensor voting [15]. By looking for local maxima, the number of new clusters and their centers are detected.

Using these results for initialization, the EM algorithm is applied to estimate the cluster parameters. The probability that a feature point i originates from a cluster m can be estimated from its location and velocity, and is defined as

$$f_m(i) \propto \mathrm{e}^{-\operatorname{dist}(x_i, \lambda_m)} \tag{1}$$

where the distance between a point and a cluster is

$$\operatorname{dist}(x_i, \lambda_m) = \begin{bmatrix} u_i - o_m \\ s_i - v_m \end{bmatrix}^T \begin{bmatrix} \Sigma_m^o & \\ & \Sigma_m^v \end{bmatrix}^{-1} \begin{bmatrix} u_i - o_m \\ s_i - v_m \end{bmatrix}. \tag{2}$$

According to Bayes' theorem, the posterior probability that point i is generated by cluster m is $p_m(i) = \frac{w_m f_m(i)}{\Sigma w_m f_m(i)}$, where w_m is the prior probability of cluster m defined as the fraction of image pixels generated from it. Points are associated with the cluster that maximizes the posterior probability. Once all the points are assigned, the parameters of each cluster are re-estimated by summing the evidence over all its points. This is iterated until EM converges to a local maximum of the likelihood of the observed data. A phase of K-Means clustering is inserted to obtain a better initialization so that the EM algorithm converges with fewer iterations. In fact, in cases where objects are well separated, EM does not change the output of K-Means at all. Results are shown in Figure 2 .

4. Mixture Particle Filtering

4.1. Initialization of Individual Particle Filters

Given the initial parameters of a cluster obtained from the cluster analysis step, a particle filter is instantiated. Two sets of particles are sampled in each filter, one from the initial distribution of the cluster estimated by the EM algorithm, $x_{i,0}'^m = \operatorname{sample}(\lambda_0^m)$, and the other around each feature point in the cluster, $x_{i,0}'^m = x_{j,0}^m + [\varepsilon_u, \varepsilon_s]^T$, where ε_u and ε_s are random variables modeling respectively the changes in space and in velocity, shown in Figure 2 .

Particles sampled from the cluster distribution can fill the gaps of particles sampled around feature points so that the distribution of the cluster is fully and well approximated. In all experiments, 100 particles are sampled around a feature

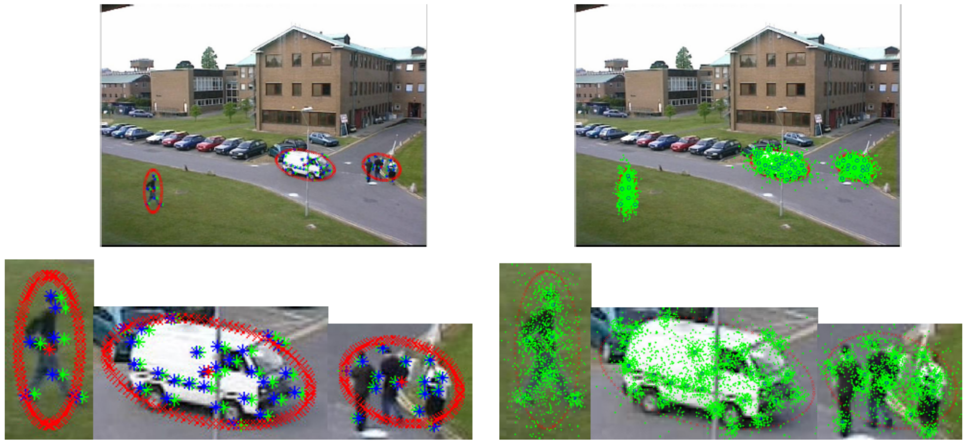

Figure 2: Results of EM-based cluster analysis and initialization of individual particle filters.

point, and the number of particles sampled from the cluster distribution is proportional to the size of the cluster.

4.2. Tracking by MPF

In MPF, tracking maintains and propagates the mixture distribution in time. For our specific problem, we also need to assign feature points in the new frame to a cluster.

A particle in filter m is propagated in the sequence based on the constant velocity assumption

$$x'^m_{i,t+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x'^m_{i,t} + \begin{bmatrix} \varepsilon_u \\ \varepsilon_s \end{bmatrix}, \tag{3}$$

and weighted by a function of the distances from the observations, feature points, defined as

$$w^m_{i,t+1} \propto \sum_j \mathrm{e}^{-\operatorname{dist}(x'^m_{i,t+1}, x_{j,t+1})}, \tag{4}$$

where the distance is

$$\operatorname{dist}(x'^m_{i,t+1}, x_{j,t+1}) = \tag{5}$$
$$(x'^m_{i,t+1} - x_{j,t+1})^T \begin{bmatrix} \Sigma_u^{-1} & \\ & \Sigma_s^{-1} \end{bmatrix} (x'^m_{i,t+1} - x_{j,t+1}).$$

Σ_u and Σ_s are set to balance the influence of the distance in space and in velocity. Thus, the mixture distribution is updated, and so are the parameters of the clusters.

However, the weight computed by Equation 4 only reflects the similarity of the particle with its neighboring feature points. It is possible that a "bad" particle is enhanced by feature points in another cluster, which happens during occlusion. To overcome this problem, the particles are reweighted using the information of the current cluster parameters

$$w^m_{i,t+1} \propto (1-\alpha) \sum_j \mathrm{e}^{-\operatorname{dist}(x'^m_{i,t+1}, x_{j,t+1})} + \alpha\, \mathrm{e}^{-\operatorname{dist}(x'^m_{i,t+1}, \lambda^m_{t+1})}. \tag{6}$$

The second term, which is the probability that the particle originates from a cluster defined by Equation 1, is added to penalize the similarity measurement of the particle with the cluster.

Feature points are then clustered according to the current cluster parameters. As stated above, feature points sometimes disappear due to occlusion or deformation, and new feature points arise. Therefore, new particles are sampled around each feature point and are weighted using Equation 5. Together with existing particles, both the mixture distribution and the cluster parameters are refined. As did Vermaak *et al.* [6], the component weight of a cluster $\pi_{m,t+1}$ is updated by summing the weights of its particles and normalizing among all the clusters, which is the only place where filters interact.

The final step of a particle filter is to resample particles based on their weights so that particles with small weights are likely to be discarded and those with large weights are duplicated. Note that a fixed number of particles in a filter are selected during tracking.

In summary, the MPF based tracker consists of the following steps: (1) Prediction: particles are propagated using Equation 3. (2) Weighting: they are weighted using Equation 4. (3) Clustering: assign feature points in the current frame to a cluster, new particles are sampled around each feature point. (4) Reweighting: particles are reevaluated using Equation 6, and the component weights of clusters are computed. (5) Resampling: resample particles using the Monte Carlo Sampling technique. These steps are iterated to propagate the mixture distribution in time.

4.3. Which Clusters to Track?

One key issue in the MPF is to determine the correct number of components (clusters in our case) that are present in the mixture. Individual filters are initialized in the Clustering step when a large number of feature points exist that are not associated with any existing filters. At the Weighting step, if the component weight of a filter drops below a threshold, the filter will be terminated. This happens when the target is occluded e.g. by foreground objects, or leaves the scene.

Occlusion among targets is solved by merging overlapping clusters and splitting widely dispersed clusters. Basically, a test of merging is performed when two clusters i and j overlap or when their centers are close to each other. The mixture distribution of the merged cluster k is estimated from the individual distributions. The cost of merging clusters C_t^i and C_t^j into C_t^k is defined as

$$\mathrm{cost}(C_t^i, C_t^j, C_t^k) = (1-\beta)\frac{|\Sigma_{k,t}^o|}{|\Sigma_{i,t}^o| + |\Sigma_{j,t}^o|} + \beta\frac{|\Sigma_{k,t}^v|}{|\Sigma_{i,t}^v| + |\Sigma_{j,t}^v|}.$$

When this cost is smaller than a predefined threshold, meaning that the clusters overlap not only in space but also in velocity, the merge is accepted.

Likewise, when a cluster grows substantially, a test of splitting is performed using the same cost function, except that the split is accepted only if the cost is larger than a predefined threshold. Note that a conservative threshold should be set for merging and splitting in order to keep the number of the clusters stable.

5. Results

The proposed method was evaluated on different sequences from PETS2001. Figure 3 shows the result of tracking two crossing targets in a short sequence of 65 frames. Thanks to the use of motion coherence, they are tracked separately without being merged during occlusion. In Figure 4, results of tracking two sequences of 300 frames taken from the same scene but at different views are shown. Challenging sequences were also chosen to evaluate the robustness of the method, as is demonstrated in Figure 5. The first sequence, shown in the left panel of Figure 5, contains large illumination changes and a complete occlusions. As expected, the algorithm proves robust to illumination changes but incapable of handling the occlusion. During the occlusion, the two filters are merged and then split as new targets. The second sequence, shown in the middle panel, contains large and strong shadows. As shadows have a motion similar to the targets who cast them, they are tracked as a part of the targets and introduce only little jitter in the trajectories. The last sequence, shown in the right panel, is the hardest which contains illumination changes, shadows, severe occlusions and numerous groups of people entering and leaving. The algorithm has problems to maintain a correct number of clusters, because shadows connect distinct clusters and people move from one cluster to another. However, an advantage of our method is that the errors are not propagated in the sequence so that interactive reinitialization is unnecessary. Note that the sequences are very noisy and only long trajectories with large certainties are displayed.

6. Conclusion and Future Work

This paper presents a novel method of tracking moving targets via feature points. The key of the method is to propagate the mixture point distribution in time using the mixture particle filter. The EM algorithm is applied to start and initialize each individual filter. Filters are terminated, merged and split during occlusion, targets entering and leaving the scene. As demonstrated, the method is robust and capable of dealing with partial occlusions. In case of total occlusion, new clusters are detected and are not linked to their correspondences before occlusion due to the lack of other information such as the appearance of the targets.

We are currently focusing on tracking in more difficult situations such as large illumination changes, shadows and severe occlusions. Complementary methods for tracking individual targets over long sequences are being developed using model-based approaches and probabilistic data association. An extension of the current work to moving cameras is also ongoing and will broaden its application to other fields, for instance, sports analysis.

References

[1] Hayet, J.B., Piater, J., Verly, J.: Robust incremental rectification of sports video sequences. In Proc. of the British Machine Vision Conference (BMVC), 2004.

[2] Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization method. Int. J. of Comp. Vision, 9(2):137–154, 1992.

[3] Loutas, E., Diamantaras, K., Pitas, I., Occlusion Resistant Object Tracking, ICIP01, vol. 2, pp. 65-68, 2001.

[4] McLachlan, G.J., Krishnan, T.: The EM Algorithm and Extensions, Wiley, New York, 1997.

[5] Arulampalam, S., Maskell, S., Gordon, N., Clap, T.: A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking, IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 174-188, 2002.

[6] Vermaak, J., Doucet, A., Perez, P.: Maintaining multimodality through mixture tracking, International Conference on Computer Vision 2003, Nice, France, 2003.

[7] Okuma, K., Taleghani, A., Freitas, N.D., Little, J.J., Lowe, D.G.: A boosted particle filter: multitarget detection and tracking, ECCV2004, vol.1, pp. 28-39, Prague, Czech, 2004.

[8] Harris, C., Stephens, M.: A combined corner and edge detector, Fourth Alvey Vision Conference, pp. 147-151, 1988.

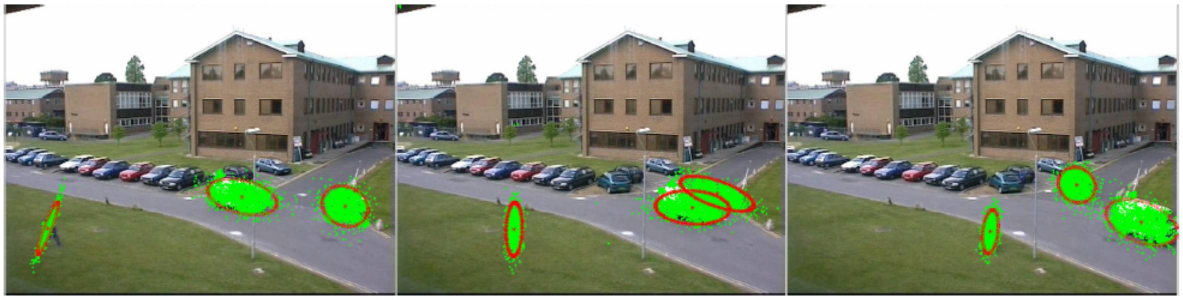

Figure 3: Results of tracking under partial occlusion. Green dots are sampled particles.

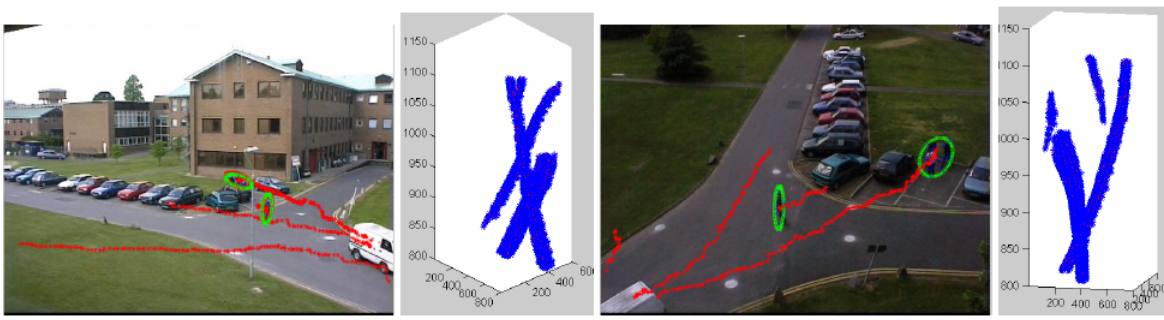

Figure 4: Results of tracking two sequences taken from the same scene with different views. Sampled particles are displayed in the spatio-temporal space.

Figure 5: Results of tracking 3 subsequences clipped from a long sequence in PETS2001.

[9] Lucas, D.B., Kanade, T.: An iterative image registration technique with an application to stereo vision, International Joint Conference on Artificial Intelligence, pp. 674-679, 1981.

[10] Ferryman, J., Clark, A., Crowley, J.L., eds.: Proceedings of the Second IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, IEEE Computer Society, 2001.

[11] Khurram Shafique, Mubarak Shah: A noniterative greedy algorithm for multiframe point correspondence, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 1, January 2005.

[12] Arnaud, E., Memin, E., Cernuschi-Frias, B.: A robust stochastic filter for point tracking in image sequences. Asian Conference on Computer Vision 2004, Korea.

[13] Wieghardt, J., Würtz, R.P., Malsburg, C.: Gabor-based feature point tracking with automatically learned constraints. Dynamic Perception, pp. 121-126, infix/IOS Press, 2002.

[14] Pece, A.E.C.: Generative-model-based tracking by cluster analysis of image differences, Robotics and Autonomous Systems, vol. 49(3-4), pp. 181-194, 2002.

[15] Medioni, G., Tang, C.K.: Inference of integrated surface, curve and junction descriptions from sparse 3-D data, IEEE Transactions on PAMI, vol. 20, no. 11, pp. 1206-1223, November 1998.