MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Covariance Tracking using Model Update Based on Lie Algebra

Fatih Porikli, Oncel Tuzel

TR2005-127   June 2006

Abstract

We propose a simple and elegant algorithm to track nonrigid objects using a covariance based object description and a Lie algebra based update mechanism. We represent an object window as the covariance matrix of features, therefore we manage to capture the spatial and statistical properties as well as their correlation within the same representation. The covariance matrix enables efficient fusion of different types of features and modalities, and its dimensionality is small. We incorporated a model update algorithm using the Lie group structure of the positive definite matrices. The update mechanism effectively adapts to the undergoing object deformations and appearance changes. The covariance tracking method does not make any assumption on the measurement noise and the motion of the tracked objects, and provides the global optimal solution. We show that it is capable of accurately detecting the nonrigid, moving objects in non-stationary camera sequences while achieving a promising detection rate of 97.4 percent.

IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)

Covariance Tracking using Model Update Based on Lie Algebra

Fatih Porikli[§] Oncel Tuzel[†§] Peter Meer[†‡]
[§]Mitsubishi Electric Research Laboratories
Cambridge, MA 02139
[†]CS Department & [‡]ECE Department
Rutgers University, Piscataway, NJ 08854

Abstract

We propose a simple and elegant algorithm to track non-rigid objects using a covariance based object description and a Lie algebra based update mechanism. We represent an object window as the covariance matrix of features, therefore we manage to capture the spatial and statistical properties as well as their correlation within the same representation. The covariance matrix enables efficient fusion of different types of features and modalities, and its dimensionality is small. We incorporated a model update algorithm using the Lie group structure of the positive definite matrices. The update mechanism effectively adapts to the undergoing object deformations and appearance changes. The covariance tracking method does not make any assumption on the measurement noise and the motion of the tracked objects, and provides the global optimal solution. We show that it is capable of accurately detecting the non-rigid, moving objects in non-stationary camera sequences while achieving a promising detection rate of 97.4 *percent.*

1. Motivation

Finding the correspondences of the previously detected objects in the current frame, tracking, is an essential component of several vision applications. Still, robust and accurate tracking of a deforming, non-rigid and fast moving object without getting restricted to particular model assumptions presents a major challenge.

Here we briefly describe the conventional tracking methods and their latent shortcomings. Mean-shift [5] is a non-parametric density gradient estimator to find the image window that is most similar to the object's color histogram in the current frame. It iteratively carries out a kernel based search starting at the previous location of the object. Even though there are variants [11] to improve its localization by using additional modalities, the original method requires the object kernels in the consecutive frames to have a certain overlap. The success of the mean-shift highly depends on the discriminating power of the histograms that are considered as the objects' probability density function.

Tracking can be considered as estimation of the state given all the measurements up to that moment, or equivalently constructing the probability density function of object location. A common approach is to employ predictive filtering and use the statistics of object's color and location in the distance computation while updating the object model by constant weights [15]. When the measurement noise are assumed to be Gaussian, the optimal solution is provided by the Kalman filter [3]. When the state space is discrete and consists of a finite number of states, Markovian filters can be applied for tracking. The most general class of filters is represented by particle filters, which are based on Monte Carlo integration methods. The current density of the state (which can be location, size, speed, boundary [9], etc.) is represented by a set of random samples with associated weights and the new density is computed based on these samples and weights. Particle filtering is a popular tracking method [2],[16], [4]. However, it is based on random sampling that becomes a problematic issue due to sample degeneracy and impoverishment, especially for higher dimensional representations.

Tracking can also be considered as a classification problem and a classifier can be trained to distinguish the object from the background [1]. This is done by constructing a feature vector for every pixel in the reference image and training a classifier to separate pixels that belong to the object from pixels that belong to the background. As in the mean-shift, an object can be tracked only if its motion is small. One obvious drawback of the local search methods is that they tend to stuck into the local optimum.

A major concern is the lack of a competent similarity criterion that captures both statistical and spatial properties, i.e., most approaches either depend only on the color distributions or structural models. Many different representations, from aggregated statistics to appearance models, have been used for tracking objects. Color histograms are popular representations of nonparametric density, but they disregard the spatial arrangement of the feature values. Moreover, they do not scale to higher dimensions due to exponential size and sparsity. Appearance models map the image features onto a fixed size window. Since the dimensionality

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

is a polynomial in the number of features and the window size, only a relatively small number of features can be used. Appearance models are highly sensitive to the pose, scale and shape variations.

To overcome the shortcomings of the existing approaches, we proposed a covariance matrix representation to describe the object windows. We generalize the idea presented in [13] to tracking problems. In the next section, we explain how we construct the covariance matrices, compute the distances, and update the models. In Section 3, we give several examples of non-rigid object tracking under varying illumination conditions, and fusion of infrared and color information.

2. Covariance Tracking

A brief description of the tracking algorithm is as follows. At each frame, we construct a feature image (Section 2.1). For a given object region, we compute the covariance matrix of the features as the model of the object (Section 2.2). In the current frame, we find the region that has the minimum covariance distance from the model and assign it as the estimated location (Section 2.3). To adapt to variations, we keep a set of previous covariance matrices and extract an intrinsic mean using Lie algebra (Section 2.4).

2.1. Features and Spatial Arrangements

We denote the observed image with I, where it might be one dimensional intensity image or three dimensional color image, or four dimensional combination of color and infrared images, or etc. Let F be the $W \times H \times d$ dimensional feature image extracted from I

$$F(x, y) = \Phi(I, x, y)$$

where the function Φ can be any mapping such as color, image gradients $I_x, I_{xx}, ..$, edge magnitude, edge orientation, filter responses, etc. This list can be extended by including higher order derivatives, texture scores, and temporal frame differences, etc. For a given rectangular window $R \subset F$, let $\{\mathbf{f}_k\}_{k=1..n}$ be the d-dimensional feature vectors inside R. We construct the feature vector $\mathbf{f}_k$ using two types of mappings; spatial attributes that are obtained from pixel coordinate values, and appearance attributes, i.e., color, gradient, infrared, etc. These features may be associated directly to the pixel coordinates

$$\mathbf{f}_k \quad = \quad [\, x \quad y \quad I(x,y) \quad I_x(x,y) \quad ... \,].$$

Alternatively, they can be arranged in radially symmetric relationship

$$\mathbf{f^r}_k \quad = \quad [\; ||(x', y')|| \quad I(x,y) \quad I_x(x,y) \quad ... \;]$$

where

$$||(x', y')|| = \sqrt{(x'^2 + y'^2)}, \quad (x', y') = (x - x_0, y - y_0)$$

are the relative coordinates, and (x_0, y_0) is the coordinates of the window center.

Different associations of the spatial information to the image features enables imposing of separate blending rules. For instance, $\mathbf{f}_k$ prevails an appearance model susceptible to the object rotation with respect to window origin (x_0, y_0), whereas $\mathbf{f^r}_k$ offers rotation invariant spatial formation of the features.

2.2. Covariance Matrix

We represent an $M \times N$ rectangular region R with a $d \times d$ covariance matrix $\mathbf{C}_R$ of the feature points as

$$\mathbf{C}_R = \frac{1}{MN} \sum_{k=1}^{MN} (\mathbf{f}_k - \boldsymbol{\mu}_R)(\mathbf{f}_k - \boldsymbol{\mu}_R)^T \qquad (1)$$

where $\boldsymbol{\mu}_R$ is the vector of the means of the corresponding features for the points within the region R. The covariance matrix is a symmetric matrix where its diagonal entries represent the variance of each feature and the non-diagonal entries represent their respective correlations.

There are several advantages of using covariance matrices as region descriptors. The covariance matrix proposes a natural way of fusing multiple features without normalizing features or using blending weights. It embodies the information embedded within the histograms as well as the information that can be derived from the appearance models. In general, a single covariance matrix extracted from a region is enough to match the region in different views and poses. The noise corrupting individual samples are largely filtered out with the average filter during covariance computation. Covariance matrix of any region has the same size, thus it enables comparing any regions without being restricted to a constant window size. It has also an scale invariance property over the regions in different images in case the raw features such as, image gradients and orientations, are extracted according to the to scale difference.

As given above, covariance matrix can be invariant to rotations. Nevertheless, if information regarding the orientation of the points are embedded within the feature vector, it is possible to detect rotational discrepancies. We also want to point that the covariance is invariant to the mean changes such as identical shifting of color values. This becomes an advantageous property when objects are tracked under varying illumination conditions.

It is possible to compute covariance matrix from feature images in a very fast way using integral image representation [10]. After constructing tensors of integral images corresponding to each feature dimension and multiplication of any two feature dimensions, the covariance matrix of any arbitrary rectangular region can be computed independent of the region size. Refer to [13] for more details.

2.3. Finding the Best Match

To obtain the most similar region to the given object, we need to compute distances between the covariance matrices corresponding to the target object window and the candidate regions. However, the covariance matrices do not lie on the Euclidean space. For example, the space is not closed under multiplication with negative scalers. Therefore an arithmetic subtraction of two matrices would not measure the distance of the corresponding regions.

Supposing no features in the feature vector would be exactly identical, which states the covariance matrices are positive definite, it is possible apply the distance measure proposed by Förstner [7]. The distance metric uses the sum of the squared logarithms of the generalized eigenvalues to compute the dissimilarity between covariance matrices as

$$\rho(\mathbf{C}_i, \mathbf{C}_j) = \sqrt{\sum_{k=1}^{d} ln^2 \lambda_k(\mathbf{C}_i, \mathbf{C}_j)} \tag{2}$$

where $\{\lambda_k(\mathbf{C}_i, \mathbf{C}_j)\}$ are the generalized eigenvalues of $\mathbf{C}_i$ and $\mathbf{C}_j$, computed from

$$\lambda_k \mathbf{C}_i \mathbf{x}_k - \mathbf{C}_j \mathbf{x}_k = 0 \quad k = 1 \ldots d \tag{3}$$

and $\mathbf{x}_k$ are the generalized eigenvectors. The distance measure ρ satisfies the metric axioms, positivity, symmetry, triangle inequality, for positive definite symmetric matrices.

At each frame we search the whole image to find the region which has the smallest distance from the current object model. The best matching region determines the location of the object in the current frame.

2.4. Model Update Strategies

Since non-rigid and moving objects undergo shape, size, and appearance transformations in time, it is necessary to adapt to these variations. We construct and update a temporal kernel of covariance matrices corresponding to the previously estimated object regions $R_1, \ldots, R_T$. We keep a set of T previous covariance matrices $[\mathbf{C}_1 \, \ldots \, \mathbf{C}_T]$ where $\mathbf{C}_1$ denotes the current covariance matrix. From this set, we compute a sample mean covariance matrix that blends all the previous matrices.

In case all the previously detected regions and the corresponding feature measurements are stored, an aggregated covariance matrix can be obtained by

$$\widetilde{\mathbf{C}} = \begin{bmatrix} \sigma_{1,1}^2 & \sigma_{1,2}^2 & \cdots \\ \sigma_{1,2}^2 & \sigma_{2,2}^2 & \\ \vdots & & \ddots \end{bmatrix}_{d \times d} \tag{4}$$

where the entries are defined as

$$\sigma_{u,v}^2 = \frac{1}{MNT} \sum_{t=1}^{T} \sum_{k=1}^{MN} \left[f_k^t(u) - \mu(u) \right] \left[f_k^t(v) - \mu(v) \right] \tag{5}$$

and $\mathbf{f}_k^t \in R_t$. The mean $\boldsymbol{\mu}$ is computed over all regions $R_1, \ldots, R_T$. Although this formulation is arguably straightforward, it assumes that all the windows have identical sizes and they are equally influential. Besides, it is computationally expensive, $\mathcal{O}(MNTd^2)$ and requires a large amount of memory to store all the previous observations.

It is desirable to obtain an aggregated covariance matrix without being limited to a constant window size and keeping all the previous measurements. We want to compute a mean covariance matrix, an *intrinsic average*. However, covariance matrices do not conform to Euclidean geometry. We can still find the mean of several covariance matrices through Riemannian geometry since symmetric positive definite matrices have Lie group structure.

Here we provide a brief overview of Lie group and algebra. A Lie group is an analytic manifold that is also a group such that the group operations

- multiplication $\quad (\mathbf{A}, \mathbf{B}) \mapsto \mathbf{AB} : G \times G \mapsto G$
- inversion $\quad \mathbf{A} \mapsto \mathbf{A}^{-1} : G \mapsto G$

are differentiable maps. Lie groups can be locally viewed as topologically equivalent to the vector space, $\mathbb{R}^d$. Thus, the local neighborhood of any group element $\mathbf{A}$ can be adequately described by its tangent space. The tangent space at the identity element of the group $\mathbf{e}$, forms a Lie algebra $\mathfrak{g}$. A Lie algebra $\mathfrak{g}$ is a vector space. Note that, we use small letters for elements of Lie algebra and capital letters for elements of Lie group. For more details on Lie groups refer to [12].

The exponential map, $\exp : \mathfrak{g} \mapsto G$, maps the vectors in the Lie algebra to the Lie group. We focus on matrix Lie groups only. The exponential map of a matrix and its inverse, $\log$, is defined by

$$\exp(\mathbf{a}) = \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{a}^n \;\; \log(\mathbf{A}) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} (\mathbf{A} - \mathbf{e})^n. \tag{6}$$

For commutative groups, the exponential map satisfies the identity $\exp(\mathbf{a})\exp(\mathbf{b}) = \exp(\mathbf{a} + \mathbf{b})$. This identity does not hold for non-commutative Lie groups such as covariance matrices. The mapping is defined by $\exp(\mathbf{a})\exp(\mathbf{b}) = \exp(\mathrm{BCH}(\mathbf{a}, \mathbf{b}))$ through Baker-Campbell-Hausdorff formula

$$\mathrm{BCH}(\mathbf{a}, \mathbf{b}) = \mathbf{a} + \mathbf{b} + \frac{1}{2}[\mathbf{a}, \mathbf{b}] + O(|(\mathbf{a}, \mathbf{b})|^3), \tag{7}$$

where $[\mathbf{a}, \mathbf{b}] : \mathfrak{g} \times \mathfrak{g} \mapsto \mathfrak{g}$ is the Lie bracket operation.

In several applications the Lie algebra is used for computing intrinsic means of points having Lie group structure [6, 8, 14]. We adapt the similar idea to obtain the intrinsic mean of covariance matrices. Let $\mathbf{c}$ be a point on the Lie algebra and $\mathbf{C} = \exp(\mathbf{c})$ be its mapping to the Lie group. The distances on manifolds are defined in terms of minimum length curves between points on the manifold. The curve with the minimum length is called the *geodesic* and the length of the curve is the *intrinsic distance*. The intrinsic distance of point $\mathbf{C}$ to the identity element of the group $\mathbf{e}$ is given by $\|\log(\mathbf{C})\|$. Left multiplication by the inverse of a group element $\mathbf{C}^{-1}$ maps the point $\mathbf{C}$ to $\mathbf{e}$ and tangent space at $\mathbf{C}$ to Lie algebra. This mapping is an isomorphism. Given $\{\mathbf{C}_t\}_{t=1...T}$ as the data points on the group, taking the log of the above mapping

$$\mathbf{c}_t = \log(\mathbf{C}^{-1}\mathbf{C}_t) \tag{8}$$

the data points are mapped to the Lie algebra and $\mathbf{C}$ to $\mathbf{0}$. Since Lie algebra is a vector space, we can compute a first order approximation to the true (intrinsic) mean of the points.

Starting at an initial matrix $\mathbf{C}_1$ and iteratively computing first order approximations to the intrinsic mean, we converge to a fixed point on the group. The algorithm is summarized as follows

```
• initialize $\widehat{\mathbf{C}} = \mathbf{C}_1$
• repeat
    – for $t = 1$ to $T$
          compute $\mathbf{c}_t = \log(\widehat{\mathbf{C}}^{-1}\mathbf{C}_t)$
    – compute $\Delta\widehat{\mathbf{C}} = \exp\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{c}_t\right)$
    – assign $\widehat{\mathbf{C}} = \widehat{\mathbf{C}}\Delta\widehat{\mathbf{C}}$
• until $\|\log(\Delta\widehat{\mathbf{C}})\| < \epsilon$
```

The error at each iteration of the algorithm can be expressed in terms of higher order terms in Baker-Campbell-Hausdorff formula (7) and the mapping (8) ensures that error is minimized. At the end of the iterations, we find the intrinsic mean and use $\widehat{\mathbf{C}}$ as the current model.

In the above formulations, we considered all the previous matrices $\mathbf{C}_1 \ldots \mathbf{C}_T$ in the set as equally influential on the result regardless of whether they are accurate or not. To prevent the model from contamination, it is possible to weight the data points proportional to its similarity to the current model. Then, the computation step on the above algorithm becomes

$$\Delta\widehat{\mathbf{C}} = \exp\left(\frac{1}{\rho^*}\sum_{t=1}^{T}\rho^{-1}(\mathbf{C}_t, \mathbf{C}^*)\mathbf{c}_t\right) \tag{9}$$

where ρ is defined in (2), $\rho^* = \sum_{t=1}^{T}\rho^{-1}(\mathbf{C}_t, \mathbf{C}^*)$ and $\mathbf{C}^*$ is the model $\widehat{\mathbf{C}}$ computed at the previous frame.

Table 1. Tracking Performance Scores

	miss/total	detection†	trials ‡
Pool Player [1]	8/92	91.4	0.0356
Running Dog [1]	9/125	92.8	0.0284
Subway [1]	4/173	97.6	0.0091
Jogging [1]	20/824	97.7	0.0096
Street-color [1]	16/180	91.1	0.0351
Street-infrared [1]	61/180	66.2	1.6376
Street-joint [1]	8/180	95.6	0.0175
Race [2]	2/692	99.7	0.0015
Crowd [3]	7/522	99.1	0.0034

Percentages of correct estimation rates†, ratio of the number of trials to get a correct estimate to the total number of total locations‡. Video size 352×288^1, 352×240^2, 440×360^3.

2.5. Complexity

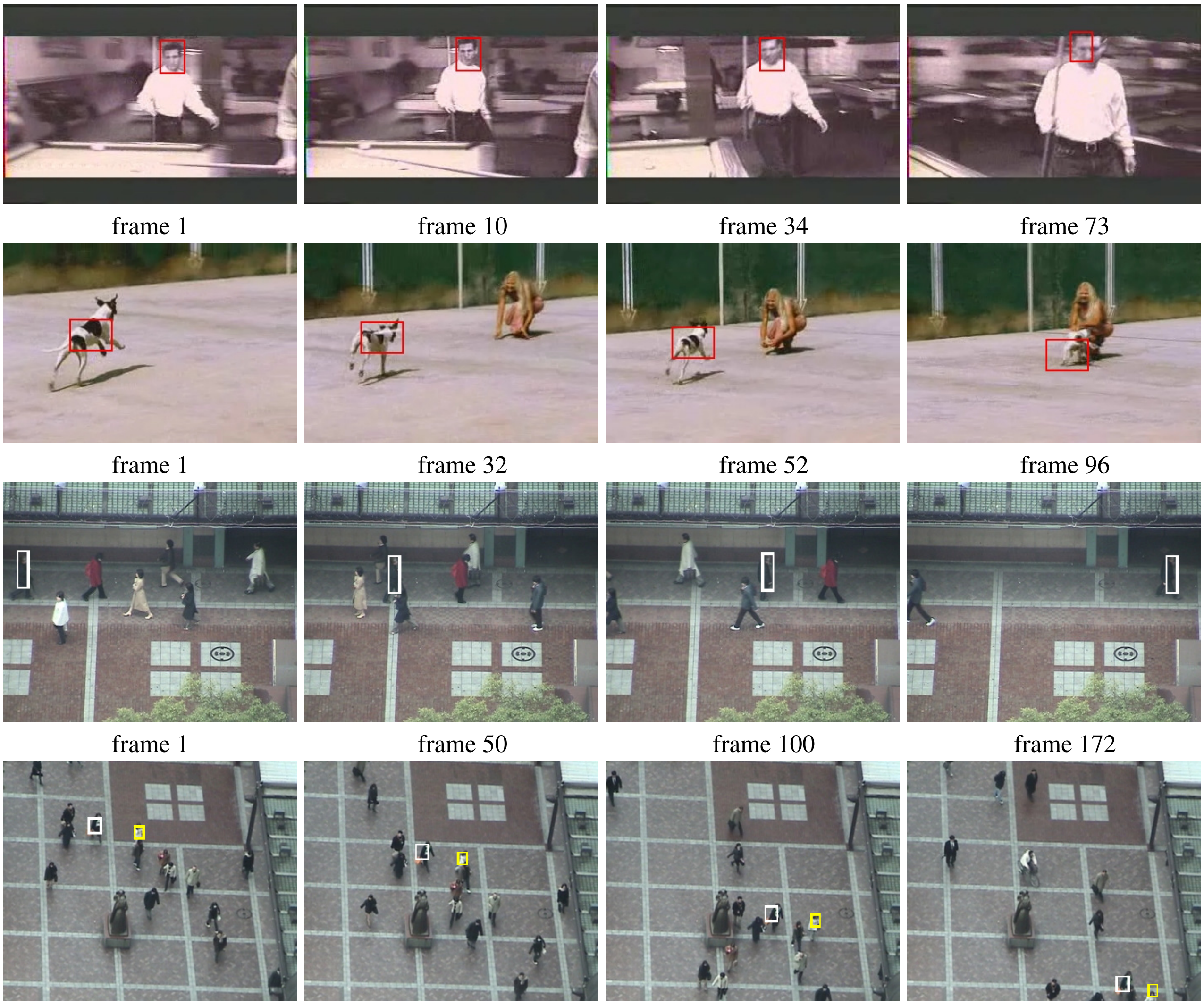
The covariance matrices are low-dimensional compared to other region descriptors and due to symmetry $\mathbf{C}_R$ has only $(d^2 + d)/2$ distinct values. Whereas if we represent the same region with the feature values we need nd values, where $n = MN$ is the number of pixels inside the object region. A conventional color histogram representation with h-quantization levels per color channel would require h^3-bins. Usually it is the case that $n \gg h \gg d$.

As mentioned before, it is possible to improve the computational complexity of covariance computation using integral histogram techniques [10, 13]. The computational complexity of constructing integral covariance representation is $\mathcal{O}(WHd^2)$. Using the constructed representation, the covariance of any rectangular region can be computed using $\mathcal{O}(d^2)$ arithmetic operations.

Most computational power is spent to compare the model with the covariance matrices of the candidate regions. The computational complexity of the distance algorithm, which requires extraction of the eigenvalues, is a polynomial in the size of the feature vector. Using the dense methods, i.e. Cholesky factorization and then QR algorithm, the complexity of distance computation can be obtained as $\mathcal{O}(d^3)$. As a result, the total complexity of the tracking becomes $\mathcal{O}(WHd^2 + WHd^3)$. For a 7×7 covariance matrix, the search takes about ~600 msec/frame on a P4 3.2GHz machine for 320×240 images. Hierarchical search methods, which are common in block matching, can also be adapted to reduce the complexity. The algorithm runs at ~150 msec/frame while achieving the same tracking performance when we apply a sampling based hierarchical search.

3. Experiments

We assessed the performance using 15 sequences totaling more than 3000 frames. These include moving and


frame 1 frame 10 frame 34 frame 73

frame 1 frame 32 frame 52 frame 96

frame 1 frame 50 frame 100 frame 172

frame 1 frame 50 frame 150 frame 250

Figure 1. Tracking results for four different sequences. In *Pool Player* and *Running Dog* sequences, the camera and objects are moving, and the appearances are changing. In *Subway* and *Crowd*, the objects have indistinctive color and insignificant texture information.

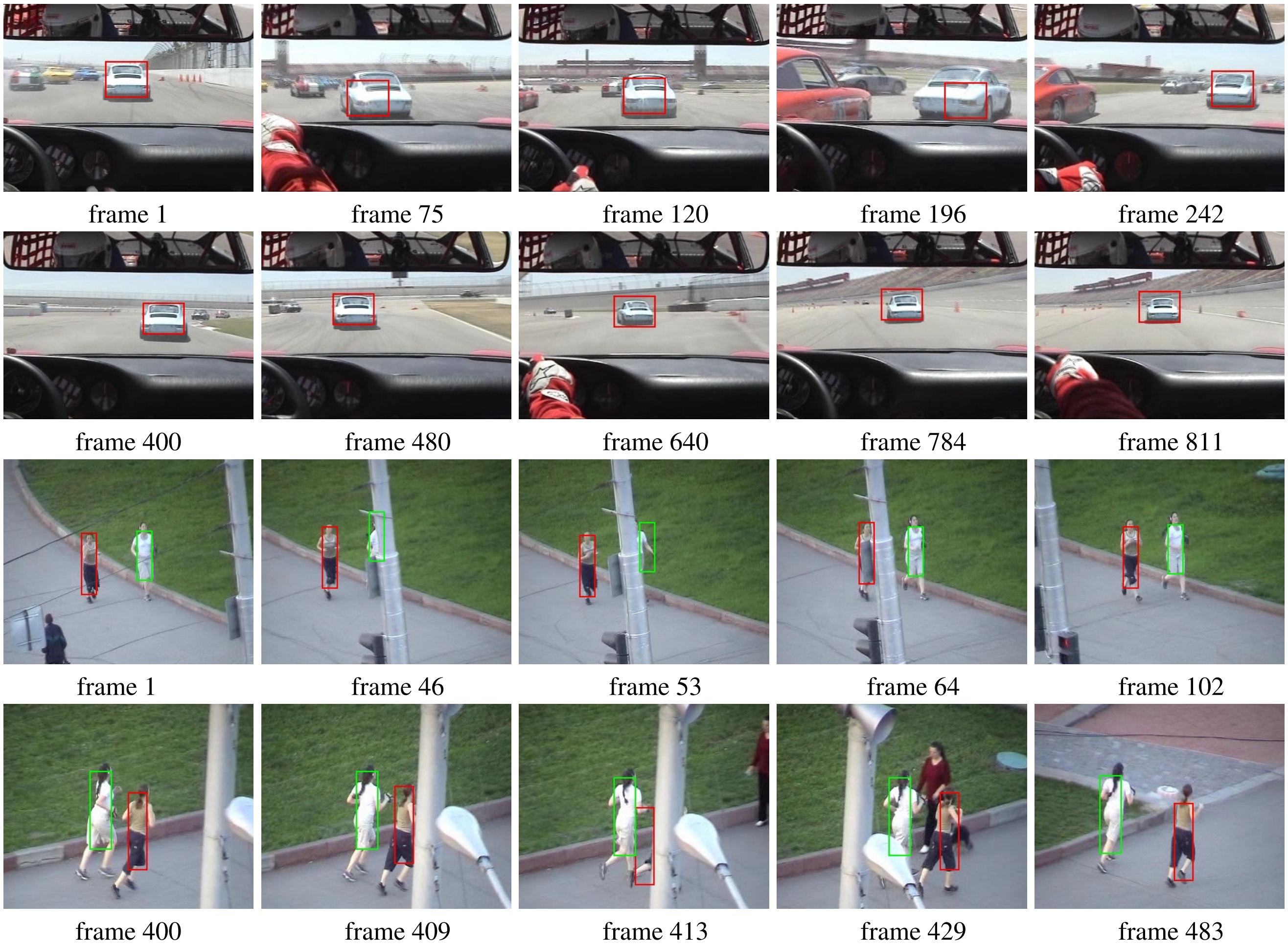
stationary camera recordings, infrared sequences, etc., and some of the results are listed in Table 1. We computed two performance metrics. The *detection rate* is the ratio of the number of frames the object location is accurately estimated to the total number of frames in the sequence. We consider the estimated location accurate if the best match is within the 9×9 neighborhood of the ground truth object center location. For example, there are 101376 possible regions for a 352×288 image and the probability of correctly estimating the object location is $1 : 1251$, if we draw it randomly.

We also analyzed the *number of trials* to find the correct estimation. This is based on ordering the search regions according to the match scores until we find the correct estimation. We defined the metric as the ratio of the total number of trials to the total number of possible regions.

Sample tracking results are given in Figures 1 and 2. For color sequences, we used all 3 RGB channels as separate features. For sequences recorded in stationary camera setups, we included a frame difference score. The frame difference feature improved the performance in infrared sequences since infrared imagery lack of sufficient spatial information to compute reliable features for small objects. For non-stationary setups, we selected the feature vector as

$$\mathbf{f}_k = [\; x \quad y \quad I(x,y) \quad |I_x(x,y)| \quad |I_y(x,y)| \;]$$

Objects are manually initialized and we applied the covariance tracking with the weighted Lie algebraic update. We computed the covariance matrices in full resolution feature image and performed the exhaustive search in half resolution grid to find the best match.



frame 1 frame 75 frame 120 frame 196 frame 242

frame 400 frame 480 frame 640 frame 784 frame 811

frame 1 frame 46 frame 53 frame 64 frame 102

frame 400 frame 409 frame 413 frame 429 frame 483

Figure 2. Tracking results using for moving camera sequences. Size changes (frames 75, 196, 242, 881) in *Race* sequence and severe occlusions (frames 53, 64, 409, 413) in *Jogging* sequence are accurately detected.

Moving Camera, Non-rigid Body: We observed that the covariance modeling and update mechanism successfully detect and adapt models to the undergoing changes as several examples are given in Figures 1 and 2. Note that, in approximately 1% of the frames the objects were fully occluded, therefore the overall detection rate was bounded at 99%. Still, the covariance tracker was able to find objects at 97.4% of the frames as given in the first column of Table 2. In comparison, optimal histogram matching could detect only 72.8% of objects in our datasets. The original mean shift [5], on the other hand, was able to keep track of objects only for a couple of initial frames in case the objects move fast and erratically (*Jogging*) or the color variation is low and object color resembles to the background (*Pool Player*). The average tracking performance of the original mean shift was less than 40%.

Table 2. Detection Rates - Gaussian Noise Contamination*

σ_η^2	0	0.01	0.1	0.3
Covariance tr.	97.4	94.3	89.0	70.6
Histogram mat.	72.8	65.2	42.6	18.9

(*) not including infrared sequences

Table 3. Detection Rates - Severe Illumination Change

	RGB	HS-only
Covariance tracking	95.6	93.3
Histogram matching	48.7	64.0

Figure 3 shows sample results with and without model update. We observed that the model update becomes more critical especially for the objects having non-rigid deformations. The model adaptation speed relies on the number of previous frames T. For example, $T = 5$ provides flexible models while $T = 40$ gives more robust estimates.

Noise and Illumination Changes: To test sensitivity against noise, we contaminated the color values with additive zero mean Gaussian noise with variance σ_η^2, where sample results are shown in Figure 4. We observed that

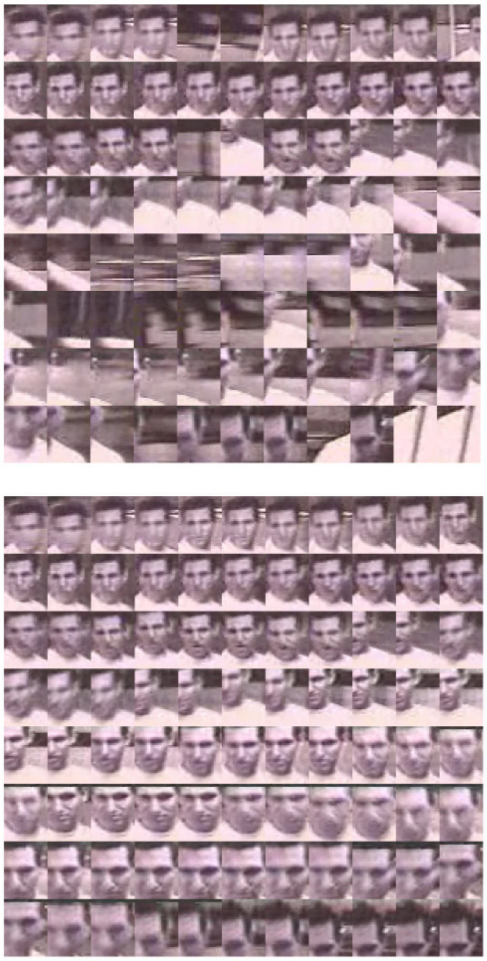


Figure 3. Montages of the detected results from 88 consecutive frames of *Pool Player* sequence. Some frames can be seen in Figure 1. With no model update; detection rate is 47.7% (top). With weighted Lie algebraic model update; detection rate is 100% (bottom).

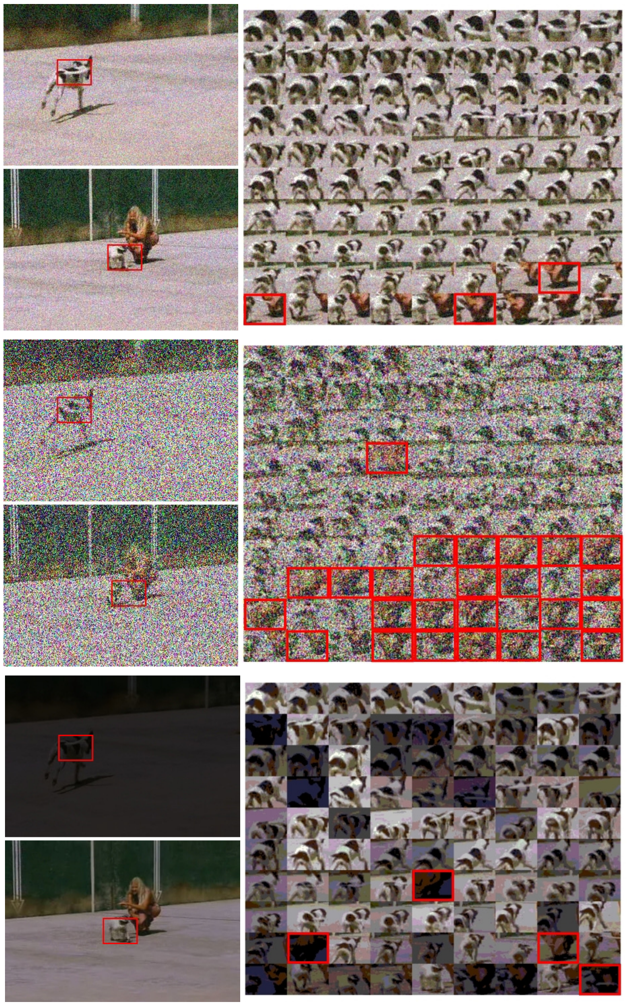


Figure 4. Frames 8 and 84 from *Running Dog* (left) and montages of 90 detected locations (right). From top to bottom: noisy data with $\sigma_\eta^2 = 0.01$ (detection rate for this sequence is 96.6%), noisy data with $\sigma_\eta^2 = 0.3$ (detection rate of 68.9%), sudden light changes (detection rate of 95.6%). Red boxes in the montage images indicate the misses.

while the performance of the histogram matching performance significantly degrades (down to 18.9%), the covariance tracking consistently achieves higher detection rates (94.3% to 70.6%), as given in Table 2. Although a common feature for tracking, histograms are easily contaminated by the noise and loose their saliency.

To analyze robustness against the illumination changes, we randomly scaled the color values of each frame as $I(x, y) = r_t I(x, y)$ where r_t is a random number between 0.2 and 1.0. The random numbers r_t, r_{t+1} were independent, thus sudden severe variations were allowed. The detection rates are given in Table 3. To be more robust toward illumination changes for histogram matching, we also tested hue-saturation values only. Still, the covariance tracking outperformed histogram matching. The last row of Figure 4 shows sample illumination transformed images and the montage images of the tracked object. The covariance tracker is very robust against the sudden illumination changes.

Fusion of Infrared and Color: The covariance matrix provides an effective solution to combine different modalities. By extending the feature vector to include the temperature values for pixel-wise aligned infrared and color sequences, we were able to take the advantage of both modalities. In Figure 5, detected objects are given for three cases; tracking using color only, infrared only, and joint as described. Detection rate has significantly improved from 92%-color and 60%-infrared to 96%-joint, and the best matches became closer to the ground truth.

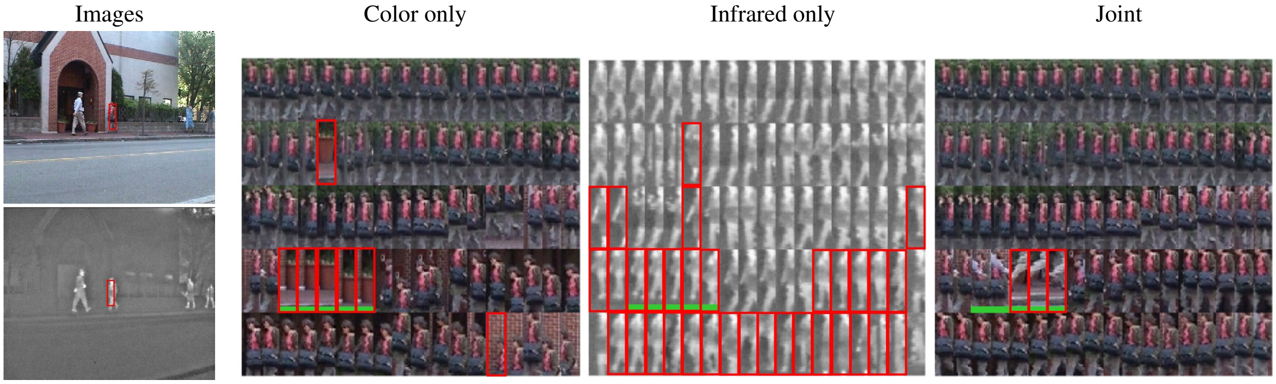


Figure 5. Detection rates of color: 92%, infrared: 60%, joint: 96%. Note that, localization also improves. Red boxes in the montage images indicate the misses. Green indicates the frames where the object is fully occluded.

4. Conclusions

We summarize the main advantages of the covariance tracking, which is a detection based localization method:

- It embodies both spatial and statistical properties of objects, and provides an elegant solution to fuse multiple features and modalities, e.g. thermal IR and color.
- It does not make any assumption on the noise and the motion model. It can track objects even if their motion is erratic and fast.
- It finds the global optimum solution unlike the local approaches such as mean shift and particle filtering.
- It can effectively adapt to temporal model changes.
- It has very low-dimensionality of $(d^2 + d)/2$.
- It is capable of comparing regions without being restricted to a constant window size.
- Our experiments show that it is robust against noise and severe lighting changes. Noise is largely filtered out during covariance computation.

References

[1] S. Avidan. Ensemble tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* San Diego, CA, 2005.

[2] N. Bouaynaya, W. Qu, and D. Schonfeld. An online motion-based particle filter for head tracking applications. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing,* Philadelphia, 2005.

[3] Y. Boykov and D. Huttenlocher. Adaptive bayesian recognition in tracking rigid objects. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Hilton Head, SC, volume II, pages 697–704, 2000.

[4] T.-J. Cham and J. M. Rehg. A multiple hypothesis approach to figure tracking. In *In Proc. Perceptual User Interfaces*, pages 19–24, 1998.

[5] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Hilton Head, SC, volume 1, pages 142–149, 2000.

[6] P. Fletcher, C. Lu, and S. Joshi. Statistics of shape via principal geodesic analysis on lie groups. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Madison, WI, volume 1, pages 95–101, 2003.

[7] W. Förstner and B. Moonen. A metric for covariance matrices. Technical report, Dept. of Geodesy and Geoinformatics, Stuttgart University, 1999.

[8] V. Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Washington, DC, volume 1, pages 684–691, 2004.

[9] M. Isard and I. Blake. Condensation – conditional density propagation for visual tracking. In *Intl. J. of Computer Vision*, volume 29, pages 5–28, 1998.

[10] F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* San Diego, CA, volume 1, pages 829 – 836, 2005.

[11] F. Porikli and O. Tuzel. Multi-kernel object tracking. In *Proceedings of IEEE Int'l. Conference on Multimedia and Expo,* Amsterdam, Netherlands, 2005.

[12] W. Rossmann. *Lie Groups: An Introduction Through Linear Groups*. Oxford Press, 2002.

[13] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *Proc. 9th European Conf. on Computer Vision,* Graz, Austria, volume 2, pages 589–600, 2006.

[14] O. Tuzel, R. Subbarao, and P. Meer. Simultaneous multiple 3d motion estimation via mode finding on lie groups. In *Proc. 10th Intl. Conf. on Computer Vision,* Beijing, China, volume I, pages 18–25, 2005.

[15] C. Wren, A. Azarbayejani, T. Darell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Anal. Machine Intell.*, 19:780–785, 1997.

[16] S. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-based modeling in particle filters. In *Proc. Intl. Conf. on Multimedia and Expo.* Baltimore, 2003.