Steepest Descent For Efficient Covariance Tracking*

Ambrish Tyagi[1] James W. Davis[1] Gerasimos Potamianos[2]
[1]Dept. of Computer Science and Engineering, Ohio State University, Columbus, OH, USA
[2]IBM T.J. Watson Research Center, Yorktown Heights, NY, USA
{tyagia,jwdavis}@cse.ohio-state.edu, gpotam@us.ibm.com

Abstract

Recent research has advocated the use of a covariance matrix of image features for tracking objects instead of the conventional histogram object representation models used in popular algorithms. In this paper we extend the covariance tracker and propose efficient algorithms with an emphasis on both improving the tracking accuracy and reducing the execution time. The algorithms are compared to a baseline covariance tracker and the popular histogram-based mean shift tracker. Quantitative evaluations on a publicly available dataset demonstrate the efficacy of the presented methods. Our algorithms obtain significant speedups factors up to 330 *while reducing the tracking errors by* $86 - 90\%$ *relative to the baseline approach.*

1. Introduction

Object tracking, which is the task of finding the association between object location from one frame to the next, has been an important research area in computer vision. The success of most practical vision applications is contingent upon obtaining high quality tracking results in real-time. The tracking problem is complicated due to several factors including the large variability in scenes (indoor vs. outdoor, lighting changes), changes in the target object (articulated, complex dynamics), and unconstrained viewing directions.

Target representation and localization methods [7, 3] for object tracking have been proposed in the literature. The authors of [7] advocate the use of covariance features over the histograms [3] to obtain a more robust algorithm, but at the same time they employ an exhaustive image search for target localization. This paper builds upon [7] and makes the following contributions: (a) We present several efficient algorithms, including a gradient descent based local optimization tracker, to improve the performance and execution time of the covariance tracker. (b) We further demonstrate the performance of these algorithms and compare them to existing approaches by performing a quantitative evaluation on a large publicly available dataset.

The remainder of the paper is organized as follows. We present the related work in Sect. 2, review the covariance tracking algorithm in Sect. 3, and describe our proposed algorithms in Sect. 4. Section 5 presents an experimental evaluation of the different tracking algorithms. Finally, we summarize and give concluding remarks in Sect. 6.

2. Related Work

A detailed survey on object tracking can be found in [10]. Tracking algorithms can be broadly classified as either filtering and data association or target representation and localization methods. Kalman and particle filters belong to the former category. For high dimensional representations, algorithms based on Monte Carlo integration (*e.g.*, particle filters) tend to be slow and become problematic due to issues of sample degeneracy and impoverishment.

Within the class of localization-based tracking algorithms mean shift tracking has gained popularity due to its computational efficiency and ease of implementation. Mean shift, a non-parametric density gradient estimator, is used to track objects by finding the mode of the similarity surface by comparing the appearance histogram of the target model with the target candidate in [3]. Similarity between the histograms is evaluated as the Bhattacharyya coefficient between the two distributions. This algorithm performs a local optimization on the search surface starting from the previously known object location.

More recently, the use of covariance features for target representation was proposed by [7]. The covariance matrix of features extracted from an image patch enables a compact representation of both the spatial and the statistical properties of the object. The tracker performs an exhaustive search in the image by comparing the given covariance model to the covariance matrix at each location using an appropriately defined distance metric. The location which is most similar to the target model is assigned to be the new target position in the image.

*This research was supported in part by the National Science Foundation under grant No. 0236653. Appears in IEEE Workshop on Motion and Video Computing (WMVC) 2008.

3. Covariance Tracking

For a given image, various features (*e.g.*, location, intensity/color, gradient) are extracted. The appearance of an image patch is compactly represented by the covariance matrix of the extracted features in that region. For tracking in a new frame, an exhaustive scan is performed to find the location that corresponds to the minimum covariance distance from the model covariance. This is the estimated new location of the object. To further improve the robustness of the algorithm, the set of previous covariance matrices are used to update the object model. We refer to this method as the full scan (FS) covariance tracker.

The covariance matrix elegantly models the joint relationship between features and their spatial distribution, unlike feature histograms used in mean shift tracking. However, the joint spatial-feature histograms can be employed in mean shift tracking [9] at the cost of obtaining high-dimensional models. It is evident that the covariance matrix representation provides a natural way of fusing multiple modalities/features without the need for normalizing the individual dimensions while keeping the dimensionality of the model low ($O(D^2)$ for D features).

Let the feature vector $\mathbf{f}_k$ at pixel location (x, y), having the color triple (r, g, b), and the x- and y- gradient values (g_x, g_y), be denoted as $\mathbf{f}_k$=$[x\ y\ r\ g\ b\ g_x\ g_y]$. For a $W \times H$ patch centered at pixel location $\mathbf{r} = (x_0, y_0)$ the covariance matrix of its features $\mathbf{f}_k$, $k = 1 \ldots WH$, is given by

$$\mathbf{C}_\mathbf{r} = \frac{1}{WH} \sum_{i=1}^{WH} (\mathbf{f}_i - \mu_\mathbf{r})(\mathbf{f}_i - \mu_\mathbf{r})^T \qquad (1)$$

where $\mu_\mathbf{r}$ is the mean feature vector for the pixels in the given image patch. The diagonal elements of the covariance matrix denote the feature variance and the off-diagonal elements correspond to their respective covariance. The $D \times D$ covariance is a symmetric positive definite (SPD) matrixand lies on a connected Riemannian manifold of Sym_D^+ [6].

Furthermore, we can normalize the $(i, j)^{th}$ element of $\mathbf{C}_\mathbf{r}$ by the product of the standard deviations of the i^{th} and the j^{th} feature dimension to obtain $\mathbf{Y}_\mathbf{r}$ (*i.e.*, the matrix of correlation coefficients). Matrix $\mathbf{Y}_\mathbf{r}$ still lies on the Sym_D^+ manifold and has the additional advantage that all values in the matrix $\in [-1, 1]$. In the rest of this paper, we employ normalized covariance matrices for tracking objects.

To compare two covariance matrices $\mathbf{M}, \mathbf{Y} \in Sym_D^+$, corresponding to the target model and the target candidate, respectively, the invariant Riemannian metric [4] can be used such that the distance between them is given by

$$\begin{aligned} d^2(\mathbf{M}, \mathbf{Y}) &= \sum_{k=1}^{D} \log^2 \lambda_k(\mathbf{M}, \mathbf{Y}) && (2) \\ &= \text{tr}\left[\log^2\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{Y}\mathbf{M}^{-\frac{1}{2}}\right)\right] && (3) \end{aligned}$$

where $\lambda_i(\mathbf{M}, \mathbf{Y})$ is the i^{th} generalized eigenvalue of $\mathbf{M}$ and $\mathbf{Y}$, or equivalently, the i^{th} eigenvalue of $\mathbf{M}^{-\frac{1}{2}}\mathbf{Y}\mathbf{M}^{-\frac{1}{2}}$, and 'tr' is the matrix trace operation.

4. Tracking Enhancements

The use of covariance features have shown to improve the tracking output in [7] where tracking was performed by scanning the entire image to find the target location. Integral histograms were used to reduce the computation and a pseudo real-time tracker was obtained that required ~ 600 msec to search a 320×240 image. This strategy of global search to find the best matching image patch (to the model covariance matrix) is slow for many real-time applications and does not scale well to larger images and higher dimensional lattices (*e.g.*, 3D tracking). Moreover, searching the entire image is prone to errors that can be caused by distractions from similar objects present in the scene.

It is not an uncommon assumption in tracking algorithms to expect that the location of the tracked object from one frame to the next changes gradually, or in other words the prior of finding the subsequent object location is stronger in the neighborhood of the previously known location [3]. We refer to this as the spatio-temporal locality assumption. In the following, we propose three optimizations based on the aforementioned assumption in order to obtain robust and computationally efficient tracking algorithms based on covariance features.

4.1. Local Search (LS)

The first apparent optimization is to limit the region of search to a local neighborhood (*e.g.*, $2W \times 2H$) centered around the previously known object location. A search strategy similar to the one presented in Sect. 3 (this time local) can be used to find the new object location by comparing the target model and candidate covariance matrices per the metric given in Eqn. 2.

4.2. Mean Shift Optimization (MS-C)

The mean shift algorithm is a non-parametric kernel density estimator that can be used for finding the modes of a distribution [2]. At each iteration, the offset ($\Delta\mathbf{x}$) between the previous location, $\mathbf{x}$, and the new kernel-weighted average of the sample points is used to find the mean shift vector that defines the path leading to a stationary point of the estimated density. The mean shift vector is given by

$$\Delta\mathbf{x} = \frac{\sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} \mathbf{y} k'(\mathbf{x}-\mathbf{y}) w(\mathbf{y})}{\sum_{\mathbf{y}\in\mathcal{N}(\mathbf{x})} k'(\mathbf{x}-\mathbf{y}) w(\mathbf{y})} - \mathbf{x}\,, \qquad (4)$$

where $k(\cdot)$ is the profile of a smooth isotropic kernel, whose derivative is $k'(\cdot)$, and $w(\cdot)$ is a weighting function. The summation is performed in the neighborhood $\mathbf{y} \in \mathcal{N}(\mathbf{x})$

of the previous location [3]. The mean shift procedure is an adaptive gradient descent method when the weights $w(\cdot)$ are constant (*i.e.*, independent of $\mathbf{x}$) [2].

The weight of a sample point can be interpreted as the influence with which it attracts the new sample mean towards itself. Hence, for the purpose of tracking, if we choose the weights appropriately such that the mean shift vector points in the direction of the search space that has the most similar appearance to the given model, then the mode of the surface will denote the new object location.

Here, we present a mean shift optimization approach to track objects using covariance features. We can use Eqn. 4 to find the target location by setting the sample weights to be equal to the inverse distance (Eqn. 2) between the model covariance matrix and the covariance matrix at the given sample point, *i.e.*, $w(\mathbf{y}) = 1/d^2(\mathbf{M}, \mathbf{Y_y})$. Using the proposed mean shift optimization still requires the calculation of the distance function (Eqn. 2) for different image locations, but a speedup will be achieved since only the weights for the pixels that fall *along* the path from current location to the density mode will be computed, as opposed to the entire neighborhood. To further speed up the search, a lookup table can be maintained to avoid re-computation of pixel weights in subsequent mean shift iterations.

4.3. Gradient Descent Optimization (GD)

It will be more beneficial in terms of computation speed if we can modify the previous tracking algorithm so that it does not require the calculation of a distance metric for various image locations. Tracking can be considered as the problem of optimizing the distance function in the search space of locations $\mathbf{x}$. A steepest/gradient descent optimization problem can be formulated which will only require the calculation of the local gradient at the current location.

In this section, we present a new approach to efficiently track objects using covariance features based on a gradient descent optimization technique. Let us reconsider the distance metric in Eqn. 3 as a function of current location $\mathbf{x}$. We are interested in minimizing the following function w.r.t. $\mathbf{x} = [x, y]^T$

$$f(\mathbf{x}) = d^2(\mathbf{M}, \mathbf{Y_x}) = \text{tr}\left[\log^2\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{Y_x}\mathbf{M}^{-\frac{1}{2}}\right)\right] \tag{5}$$

where $\mathbf{Y_x}$ is the covariance matrix at location $\mathbf{x}$. Let $P(\mathbf{x}) = \left(\mathbf{M}^{-\frac{1}{2}}\mathbf{Y_x}\mathbf{M}^{-\frac{1}{2}}\right)$. The gradient of the function $f(\mathbf{x})$ is given by

$$\nabla f(\mathbf{x}) = [\partial_x f(\mathbf{x}), \partial_y f(\mathbf{x})]^T \tag{6}$$

where,

$$\begin{aligned}
\partial_x f(\mathbf{x}) &= \partial_x d^2(\mathbf{M}, \mathbf{Y_x}) \\
&= \partial_x \text{tr}\left[\log^2 P(\mathbf{x})\right] \\
&= \text{tr}\left[\partial_x \log^2 P(\mathbf{x})\right] && (7) \\
&= \text{tr}\left[2\log P(\mathbf{x}) P(\mathbf{x})^{-1} \partial_x P(\mathbf{x})\right] && (8) \\
\partial_y f(\mathbf{x}) &= \text{tr}\left[2\log P(\mathbf{x}) P(\mathbf{x})^{-1} \partial_y P(\mathbf{x})\right] && (9)
\end{aligned}$$

such that,

$$\begin{aligned}
\partial_x P(\mathbf{x}) &= \partial_x \left(\mathbf{M}^{-\frac{1}{2}}\mathbf{Y_x}\mathbf{M}^{-\frac{1}{2}}\right) && (10) \\
&= \left(\mathbf{M}^{-\frac{1}{2}} \otimes \mathbf{M}^{-\frac{T}{2}}\right) vec(\partial_x \mathbf{Y_x}) && (11) \\
&= \mathbf{M}^{-\frac{1}{2}} \left(\partial_x \mathbf{Y_x}\right) \mathbf{M}^{-\frac{1}{2}} && (12)
\end{aligned}$$

where $\otimes$ denotes the Kronecker matrix product and $vec(\mathbf{V})$ is the vectorization operator that produces the long column vector formed by concatenating the columns of $\mathbf{V}$. The reader is referred to [5] for the derivation of Eqn. 8 from Eqn. 7.

The partials $(\partial_x \mathbf{Y_x})$ and $(\partial_y \mathbf{Y_x})$ can be estimated from discrete data by letting $dx = dy = 1$ such that

$$\begin{aligned}
\partial_x \mathbf{Y_x} &\approx (\mathbf{Y}_{\mathbf{x}|x+dx,y} - \mathbf{Y}_{\mathbf{x}|x-dx,y})/2dx && (13) \\
\partial_y \mathbf{Y_x} &\approx (\mathbf{Y}_{\mathbf{x}|x,y+dy} - \mathbf{Y}_{\mathbf{x}|x,y-dy})/2dy && (14)
\end{aligned}$$

However, since $\mathbf{Y_x} \in Sym_D^+$, the vector space subtraction operator in Eqn. 13 and Eqn. 14 should be replaced by the equivalent operations on the Riemannian manifold [6] (please refer to the Appendix)

$$\begin{aligned}
\partial_x \mathbf{Y_x} &\approx 0.5 \log_{\mathbf{Y}_{\mathbf{x}|x-dx,y}}(\mathbf{Y}_{\mathbf{x}|x+dx,y})/dx && (15) \\
\partial_y \mathbf{Y_x} &\approx 0.5 \log_{\mathbf{Y}_{\mathbf{x}|x,y-dy}}(\mathbf{Y}_{\mathbf{x}|x,y+dy})/dy && (16)
\end{aligned}$$

Therefore once we can estimate the gradient of the objective function (Eqn. 6) as described above, we can formulate a gradient descent algorithm to optimize Eqn. 5 where the new lattice location at iteration $i + 1$ is obtained as

$$\mathbf{x}^{i+1} = \mathbf{x}^i - \eta^i \nabla f(\mathbf{x}^i) \tag{17}$$

where η^i is the learning rate which follows an annealing schedule such that the i^{th} iteration learning rate is given by $\eta^i = \eta^0(1 - i/N)$, where N is determined empirically. The algorithm iterates until convergence, which occurs when $||\eta^i \nabla f(\mathbf{x}^i)|| < t_{conv}$. Finally, the model update strategy of [7] can be employed to deal with appearance changes.

5. Experiments

We now report our evaluation of the original covariance tracking algorithm (Sect. 3) and the proposed enhancements (Sect. 4) in comparison to the standard histogram-based

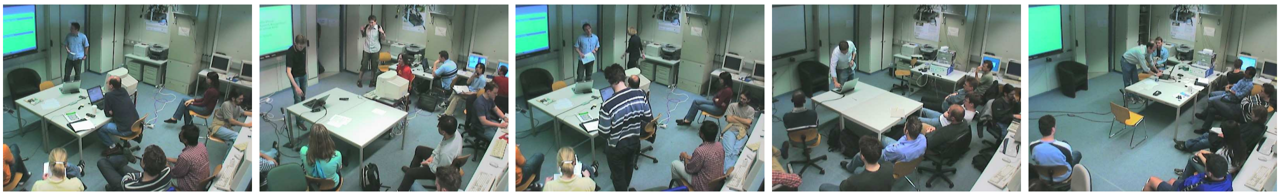


Figure 1. Representative images from the CHIL dataset. The speaker can be seen in several different poses and illumination conditions. The background is cluttered with distractions from similar objects (faces/heads).

mean shift tracker [3]. The algorithms are compared for both accuracy and computational efficiency.

For a fair comparison to the features used in covariance trackers, we used a joint histogram of color-gradient features extracted from the image for the mean shift tracker. At each pixel, a five dimensional feature (r, g, b, g_x, g_y) vector is extracted to form the $(n_{hist})^5$-dimensional quantized histogram. We refer to this tracker as MS-H.

5.1. Database and Experimental Paradigm

We evaluated the tracking algorithms on a number of "interactive seminar" video sequences recorded and annotated as part of the CHIL ("Computers in Human Interaction Loop") project [1]. The dataset consists of 22 video sequences recorded inside a smart room. Each sequence depicts a speaker giving a lecture to a small audience. Each video segment contains approximately 4500 frames recorded at 15 Hz. Images are captured at 640×480 pixel resolution. The ground truth is annotated at every 15 frames. Figure 1 shows some representative images from the dataset.

The goal of these experiments is to track the speaker's face/head through each sequence. We used the video sequences from one of the four cameras that had the best view of the main speaker's face/head. Around 100k frames were tracked in each experiment. We evaluated the various algorithms in terms of how their output differs from the annotated ground truth locations (the error is reported in pixel units).

For our experiments to be unbiased, we provided the same initial conditions for all the trackers. We initialized the trackers based on the ground truth data. In addition, tracker drift detection was achieved by comparing the tracking results to the ground truth. In particular, we introduced a drift threshold T(=30 pixels), identical to all trackers. Whenever the tracking error was higher than this predefined threshold, we re-initialized the target model for the respective tracking algorithm. The number of re-initializations required per sequence is hence another metric useful to tracking algorithm evaluation.

A superior tracking method is expected to have (1) *lower error* rates, (2) *fewer* number of *re-initializations*, and (3) *faster execution* time than competing approaches.

5.2. Performance of Various Algorithms

First we compared the tracking performance of the baseline full scan (FS) covariance algorithm (Sect. 3), local scan (LS) covariance algorithm (Sect. 4.1), mean shift covariance (MS-C) tracker (Sect. 4.2), gradient descent (GD) covariance tracker (Sect. 4.3), and the mean shift histogram (MS-H) tracker [3]. Figure 2 shows the overall tracking results (in pixel error) produced by each algorithm on the entire dataset. Tracking results averaged over different set of parameters are displayed for the MS-H, the MS-C, and the GD algorithms.

The full scan (FS) algorithm does not exploit the spatio-temporal locality assumption and hence gets distracted due to similar objects (faces) in the smart room, thus resulting in high tracking errors (100.9 pixels). The local scan (LS) algorithm on the other hand results in low tracking errors (10.4 pixels) by exploiting the above mentioned spatio-temporal locality constraint. This algorithm demonstrates good tracking performance since it performs an exhaustive search within the local neighborhood. The three optimization algorithms (MS-C, MS-H, GD) produced 14.6, 12.6, and 10.2 pixel errors, respectively. These algorithms exploit the locality constraint along with specific heuristic searching techniques (resulting in faster algorithms) to produce results comparable to the LS method. These optimization-based tracking algorithms achieve $86 - 90\%$ error reduction relative to the baseline FS method.

Within the class of optimization-based algorithms, we compared their relative tracking performance for different set of algorithmic parameters. The performance of the two mean shift trackers (MS-C and MS-H) depend on the bin quantization n_{hist} and the kernel bandwidth (or window

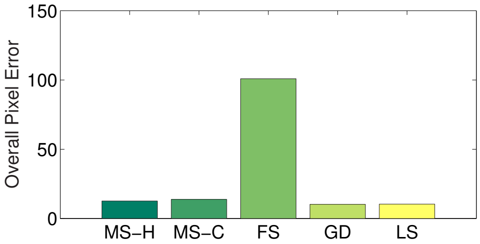
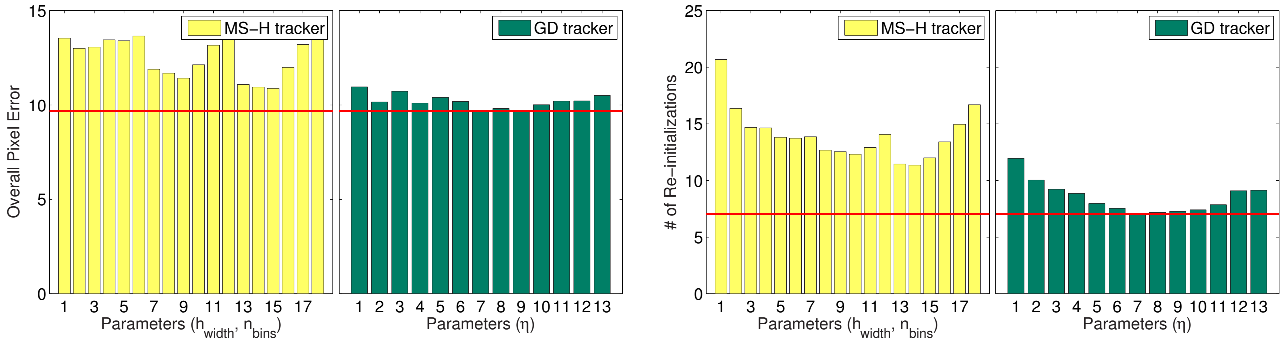


Figure 2. Overall tracking results for the various algorithms.



(a) (b)

Figure 3. Comparison of (a) the overall tracking error and (b) the number of re-initializations required by the histogram-based mean shift (MS-H) tracker and the gradient descent (GD) covariance tracker for different parameters values. The x-axis labels correspond to various parameter combinations and not their respective values.

size) h_{width}. Similarly, the learning rate (η) can influence the performance of the gradient descent (GD) algorithm.

The MS-C tracker produced errors in the range of 12.91–15.82 pixels (average 14.57 pixels) for various parameter values. Since the MS-H and GD algorithms produced lower average tracking errors compared to the MS-C method, we investigated the former two trackers in more detail.

Plots in Fig. 3(a) present the overall tracking errors on the CHIL dataset for different parameter settings of the MS-H and GD trackers. The horizontal line drawn in red is provided for reference and corresponds to the minimum error of 9.69 pixels obtained by the GD covariance tracker with learning rate of $\eta = 1$. Clearly, the GD covariance tracker outperforms the MS-H tracker, irrespective of the algorithmic parameters. Errors of the GD method were in the range of $9.68 - 10.95$ pixels (average 10.21 pixels) compared to $10.88 - 15.56$ pixels (average 12.61 pixels) for MS-H algorithm. This translates to a 19% reduction in the average error relative to the MS-H algorithm.

A similar trend is seen in Fig. 3(b) that reports the number of re-initializations required by the MS-H and GD algorithms. Again, the red line, corresponding to 7.05 restarts required by the GD tracker per sequence, is drawn for reference. The MS-H algorithm needs to be restarted 14 times per sequence (*i.e.*, every 21.5 secs) in comparison to 8.5 restarts required by the GD tracker per sequences (*i.e.*, every 35.3 secs).

5.3. Speedup

Computational efficiency of a tracking algorithm is equally important to its performance. In addition to the high tracking accuracy, we desire the proposed algorithms to be computationally efficient. Comparing the execution times for different algorithms becomes difficult without having access to their respective original implementations. In order to facilitate a comparison of computational requirements between the various covariance trackers, we implemented our own version of the full scan algorithm described in [7]. The baseline full scan algorithm can now be used to calculate the speedup obtained by the enhanced algorithms. We define speedup as the ratio of the execution time of the full scan algorithm to that of the given algorithm.

Algorithm	Speedup	Execution time per frame (msec)
LS	7.31	82.04
MS-C	24.95	24.04
GD (Worst Case)	101.50	5.91
GD (Avg. Case)	137.88	4.35
GD (Best Case)	331.07	1.81

Table 1. Speedup and estimated execution time per frame for different covariance trackers. Speedup is calculated with respect to the FS algorithm (Sect. 3). Execution time per frame is normalized with respected to the 600 msec scanning speed (for 320 × 240 images) obtained by [7] using the FS algorithm.

Moreover, [7] claims to obtain tracking times of ~ 600 msec per frame for 320×240 images using various optimizations including integral histograms. Using this information and the relative speedup factors we can estimate the running times per frame for each covariance-based tracking algorithm . The execution time per 320×240 frame is given by (600÷speedup) msec.

Table 1 reports the speedup factors (relative to the FS algorithm) and the estimated execution times per frame obtained by other covariance tracking algorithms. We report the best, worst, and average case speedups obtained by the GD algorithms for different parameter settings. The proposed enhancements to the covariance tracker (Sect. 4) result in significant speedups resulting in up to 330 times faster algorithms. High speedup factors translate to low execution times per frame and thus fast real-time tracking systems. Amongst the various covariance trackers proposed, the GD optimization-based tracker is the most computationally efficient method in all cases.

5.4. Discussion

Experimental results presented in this section demonstrate the inadequacy of the baseline FS covariance tracking algorithm, both in terms of performance and computation. Tracking algorithms that exploit the spatio-temporal locality assumption have demonstrated improved tracking accuracy. The proposed enhancements including the LS, MS-C, and the GD trackers produce lower error rates while capable of tracking objects at high frame rates.

Amongst the local search based algorithms presented in Sect. 4, the gradient descent optimization-based covariance tracker (GD) achieves the lowest tracking errors while requiring a lower number of re-initializations (Sect. 5.2) in comparison to the other covariance trackers and the standard histogram-based mean shift tracker. Moreover, the GD tracker is also very computationally efficient as can be seen from the speedup results of Sect. 5.3. The encouraging results obtained by the gradient descent method (in both accuracy and efficiency) makes it a promising real-time algorithm for tracking objects in video sequences.

5.4.1 Extension to 3D

Finally, we would like to emphasize that none of the covariance tracking algorithms, except the GD tracker, are easily scalable to 3D tracking since the number of lattice locations increase exponentially with the addition of another dimension. Nevertheless, the gradient descent algorithm can be elegantly extended to 3D tracking by optimizing the objective function (Eqn. 5) on the 3D lattice $\mathbf{X} \in [x, y, z]^T$. The covariance matrix at a lattice location $\mathbf{X}$ can be used to fuse information from multiple cameras by extracting features using the technique described in [8]. The gradient (Eqn. 6) in 3D is given by $\nabla f(\mathbf{x}) = [\partial_x f(\mathbf{x}), \partial_y f(\mathbf{x}), \partial_z f(\mathbf{x})]^T$, where $\partial_z f(\mathbf{x})$ can be obtained similarly to Eqns. 15-16. The extension of the GD tracker to 3D is a work-in-progress and will be examined in future research.

6. Summary and Conclusion

We presented efficient algorithms to track objects in video sequences using covariance features. These algorithms were compared in terms of their tracking accuracy (pixel error and number of re-initializations) and computational requirements to the recently presented covariance tracker as the baseline. A local scan version of the baseline algorithm, a mean shift covariance tracker, and a gradient descent search based covariance tracker were described. These algorithms exploit the well known spatio-temporal locality assumption, often used in tracking algorithms, to improve tracking results and reduce execution times. A relative reduction of $86-90\%$ in error and speedup factors up to 330 were obtained in comparison to the baseline. In conclusion we recommend the gradient descent algorithm to be the tracker of choice since it provides a good combination of accuracy and efficiency. This algorithm is also found to be superior when compared to the popular histogram based mean shift tracker. In future work, we plan to extend the proposed GD tracker to higher spatial (3D) dimensions and evaluate its performance.

References

[1] The CHIL consortium web-site. `http://chil.server.de`.

[2] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 24(5):603–619, 2002.

[3] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Patt. Analy. and Mach. Intell.*, 25(5):564–577, May 2003.

[4] W. Förstner and B. Mooner. A metric for covariance matrices. Technical report, department of geodesy and geoinformatics, Stuttgart University, 1999.

[5] M. Moakher. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. *SIAM J. Matrix Anal. Appl.*, 26(3):735–747, 2005.

[6] X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *Int. J. of Comp. Vis.*, 66(1):41–66, 2006.

[7] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *Proc. Comp. Vis. and Pattern Rec.*, pages 728–735, 2006.

[8] A. Tyagi, G. Potamianos, J. Davis, and S. Chu. Fusion of multiple camera views for kernel-based 3D tracking. In *Proc. Wkshp. Motion and Video Computing*, pages 1–8, 2007.

[9] C. Yang, R. Duraiswami, and L. Davis. Efficient mean-shift tracking via a new similarity measure. In *Proc. Comp. Vis. and Pattern Rec.*, pages 176–183, 2005.

[10] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), 2006.

Appendix: Manifold Operations

On the Sym_D^+ manifold, the exponential and logarithmic maps are respectively given by [6],

$$\begin{aligned} \exp_{\mathbf{Y}}(\boldsymbol{\Delta}) &= \mathbf{Y}^{1/2} \exp(\mathbf{Y}^{-1/2} \boldsymbol{\Delta} \mathbf{Y}^{-1/2}) \mathbf{Y}^{1/2} && (18) \\ \log_{\mathbf{Y}}(\mathbf{X}) &= \mathbf{Y}^{1/2} \log(\mathbf{Y}^{-1/2} \mathbf{X} \mathbf{Y}^{-1/2}) \mathbf{Y}^{1/2} && (19) \end{aligned}$$

for $\mathbf{X}, \mathbf{Y} \in Sym_D^+$ and the tangent vector $\Delta \in Sym_D$. The exp and log on r.h.s. of the above equations denote the standard matrix operations. The manifold exponential operator (Eqn. 18) is defined on the space of all symmetric matrices (not only SPD) and maps the tangent vector $\boldsymbol{\Delta}$ at $\mathbf{Y}$ to the location on the manifold reached in a unit time by the geodesic starting at $\mathbf{Y}$ in the tangent direction. Its inverse, the logarithmic operator (Eqn. 19) maps the geodesic from $\mathbf{Y}$ to $\mathbf{X}$ to the equivalent tangent vector (with the smallest norm) at $\mathbf{Y}$.