



# A visual object tracking benchmark for cell motility in time-lapse imaging

H. Seckin Demir<sup>1</sup> · A. Enis Cetin<sup>2</sup> · Rengul Cetin Atalay<sup>3</sup>

Received: 2 September 2018 / Revised: 6 December 2018 / Accepted: 2 February 2019 / Published online: 22 February 2019  
© Springer-Verlag London Ltd., part of Springer Nature 2019

## Abstract

Automatic tracking of cells is a widely studied problem in various biomedical applications. Although there are numerous approaches for the video object tracking task in different contexts, the performance of these methods depends on many factors regarding the specific application they are used for. This paper presents a comparative study that specifically targets cell tracking problem and compares performance behavior of the recent algorithms. We propose a framework for the performance evaluation of the tracking algorithms and compare several state-of-the-art object tracking approaches on an extensive time-lapse inverted microscopy dataset. We report the quantitative evaluations of the algorithms based on success rate and precision performance metrics.

**Keywords** Visual object tracking · Cell motility · Tracking benchmark

## 1 Introduction

Visual object tracking task has been an important part of different application fields such as human–computer interaction [1,2], autonomous driving [3,4], surveillance systems [5] and computerized assistance systems in medical image processing [6–9]. Although there are various approaches proposed for the visual object tracking problem [10], they generally focus on the applications in more general contexts. Therefore, datasets used for performance evaluation of the algorithms usually contain various objects or environments that can represent a wide range of scenarios. However, for the cell tracking task, the problem lies on a more restricted domain. Therefore, tracking algorithms should be analyzed and compared under a framework that contains related datasets.

Visual object tracking task plays a key role in dynamic cell behavior studies where the migration analysis of cell populations has a significant place [11]. Cell migration is

a fundamental process in regular tissue development and recovery [12]. Speed, direction and morphological changes of cells are closely related to the structure of the environment [13]. In order to move through extracellular spaces or over the surfaces of other cells, special mechanisms are employed by the individual cells [14]. These motility patterns are investigated using the microscopic image sequences in a wide range of cell types with different morphological properties. Some of the applications include red blood cell speed measurement [15], cancer cell tracking [16], Bovine Pulmonary Artery Endothelial (BPAE) cell motility tracking [17], leukocytes tracking [18,19] and embryo cell tracking [17].

In recent years, different approaches have been proposed for such analyses [20,21]. Benchmarks for comparing various methods exist for fluorescent microscopy [9,22]. However, very few studies focus on analyzing images taken by differential interference contrast (DIC), phase contrast or other label-free microscopies, which are used commonly for observing living cells [23]. In this study, we use DIC microscopy images for comparing the cell tracking performance of our algorithm with other state-of-the-art trackers.

Label-free microscopic images (especially in DIC microscopy) are usually have low-contrast gray-scale images with deformable cell shapes. Due to this property of DIC microscopy, automatic cell tracking becomes a harder task. In addition to low contrast, there exist several other challenges in cell tracking. For example, similar morphological

✉ Rengul Cetin Atalay  
rengul@metu.edu.tr

<sup>1</sup> Microelectronics, Guidance and Electro-Optics Business Sector, Aselsan, 06750 Ankara, Turkey

<sup>2</sup> Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607, USA

<sup>3</sup> Cancer Systems Biology Laboratory, Graduate School of Informatics, METU, 06800 Ankara, Turkey

structure of the cells makes it difficult to differentiate one cell from another in dense scenes. Furthermore, shape deformations and random rotations during the cell motion require adaptive models which are robust to these changes.

In this study, we present a comparative study which evaluates the robustness of our algorithm against these specific challenges in cell tracking scenarios.

In this study, we firstly established ground truth data of various cell motility image sequences. Then, we generated the tracking results of several state-of-the-art algorithms that are published in recent years. The performance of the algorithms is compared based on two different metrics that is used in [10].

The remainder of the paper is organized as follows. We briefly explain the compared tracking algorithms in Sect. 2. Then, we present the dataset, experiments and comparison results in Sect. 3. We conclude with the final remarks in Sect. 4.

## 2 Compared algorithms

Object tracking algorithms are typically grouped in two main categories: generative and discriminative methods. Generative methods, as the name suggests, construct an appearance model for the target and search for the closest match in the next frames. In general, these approaches are preferred for their computational efficiency. Discriminative methods, on the other hand, model the object and background separately and approach the tracking problem as a classification problem. In our experiments, we included methods from both approaches. Most of the algorithms, we used have publicly available codes. For all the methods, we used default parameters suggested by the authors. Brief summary of the algorithms is given in the next subsections.

### 2.1 Co-difference-based object tracking (CODIFF)

In [24], Demir et al. proposed a visual object tracking algorithm based on co-difference features. Calculation of co-difference matrix is similar to that of covariance matrix which is used as a descriptor for various vision applications such as object detection [25], classification [26] and tracking [27]. However, co-difference matrix uses a multiplierless operator for extracting descriptors in an efficient manner. Calculating the co-difference of various features such as intensity, gradients or pixel position provides a compact matrix that represents the combination of these features.

For a given subwindow  $R$  consisting of  $N$  pixels, let  $(\mathbf{f}_k)_{k=1..n}$  be the  $d$ -dimensional feature vectors in  $R$ . Then, the covariance matrix of these features for region  $R$  is calcu-

lated as follows:

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

where  $\boldsymbol{\mu}_R$  is the  $d$ -dimensional mean vector of the features calculated in region  $R$ . Although covariance matrix provides an intuitive way to fuse information coming from different features, its computational cost is relatively high due to multiplications especially for large image patches. In [28], an efficient algorithm is proposed for calculating the “covariance-like” descriptors. The main contribution that boosts the performance is the multiplication-free calculation of the descriptor. Instead of the multiplications in covariance method, this implementation employs an operator based on additions. The new operator is defined for real numbers  $a$  and  $b$  as follows:

$$a \oplus b = \begin{cases} a + b & \text{if } a \geq 0 \text{ and } b \geq 0 \\ a - b & \text{if } a \leq 0 \text{ and } b \geq 0 \\ -a + b & \text{if } a \geq 0 \text{ and } b \leq 0 \\ -a - b & \text{if } a \leq 0 \text{ and } b \leq 0 \end{cases} \quad (2)$$

In [29], it is stated that the co-difference descriptor can be calculated up to 100 times faster than the covariance matrix depending on the processor. Using this operator, a new vector product of two vectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$  of size  $N$  is given as follows:

$$\langle \mathbf{x}_1, \mathbf{x}_2 \rangle = \sum_{i=1}^N x_1(i) \oplus x_2(i) \quad (3)$$

where  $x_k(i)$  is the  $i$ -th entry of the vector  $\mathbf{x}_k$ . Now, we can define the co-difference matrix for a region  $R$  as follows:

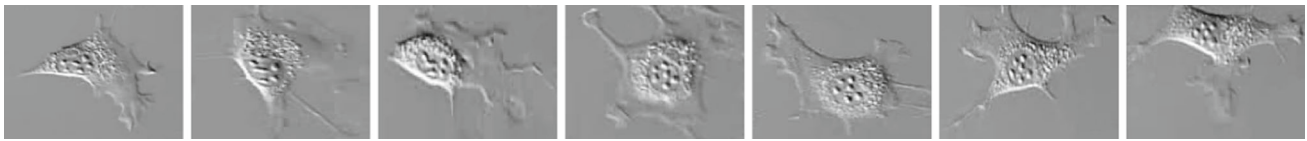
$$\mathbf{C}_d = \frac{1}{N-1} \sum_{k=1}^N (\mathbf{f}_k - \boldsymbol{\mu}_R) \oplus (\mathbf{f}_k - \boldsymbol{\mu}_R)^T \quad (4)$$

which is used as the region descriptor for visual tracking algorithm. In our comparison, we used the feature vector as follows:

$$\mathbf{f}_k = [x(k) \ y(k) \ I(k) \ I_x(k) \ I_y(k) \ I_{xx}(k) \ I_{yy}(k)] \quad (5)$$

where the elements of the feature vector are horizontal and vertical positions within the region, intensity, gradients in both directions and second derivative values in both directions, respectively. As a result, a descriptor of size  $7 \times 7$  is extracted for any given patch size.

In order to obtain the most similar region to the given object, the distances between the co-difference matrices corresponding to the target object window and the candidate



**Fig. 1** Time-lapse inverted microscopy dataset contains image sequences with challenging rotation and deformation scenarios. The shape of a sample cell is given above for frames with numbers 1, 45, 63, 97, 160, 208 and 230

regions must be computed during object tracking. This can be done by computing the generalized eigenvalues of the current matrix of the target window and the matrices of the target window. The generalized eigenvalue-based distance matrix is given by;

$$\rho(C_1, C_2) = \sqrt{\sum_i \ln^2 \lambda_i} \quad (6)$$

where  $\lambda_i$  are the generalized eigenvalues of the matrices  $C_1$  and  $C_2$ .

Although the covariance and co-difference matrices do not lie on the Euclidean space, they can be compared using the arithmetic subtraction of two matrices and computing the Euclidean norm of the difference. Since this arithmetic approach gives comparable results, Euclidean norm-based comparison is used for reducing the computational cost of the tracker.

## 2.2 Discriminative scale space tracker (DSST)

In [30], the MOSSE tracker [31] is extended with a robust scale estimation. In this method, a one-dimensional discriminative scale filter is used for estimating the target size. Another contribution of the method is employing a pixel-dense representation of HOG features in combination with the intensity features used in the MOSSE tracker for translation filter (source code available<sup>1</sup>).

## 2.3 Fast compressive tracking (FCT)

Zhang et al. used a classification-based approach in compressed domain for object tracking. In this approach, they firstly extract features from multi-scale image feature space [32]. Then, using a sparse measurement matrix, they calculate the compressed features that preserve the structure of image feature space. They use the same measurement matrix for compressing the foreground and the background samples. Thus, the tracking task is converted into a binary classification problem that will be solved with a naive Bayes classifier

<sup>1</sup> DSST Source Code: [http://www.cvl.isy.liu.se/en/research/objrec/visualtracking/scalvistrack/DSST\\_code.zip](http://www.cvl.isy.liu.se/en/research/objrec/visualtracking/scalvistrack/DSST_code.zip).

with online update in the compressed domain (source code available<sup>2</sup>).

## 2.4 Incremental learning for robust visual tracking (IVT)

In [33], Ross et al. presented a method that uses a low-dimensional subspace representation of the target object for tracking purpose. Proposed method employs an incremental PCA algorithm for adapting the appearance changes by updating the eigenbasis vectors incrementally (source code available<sup>3</sup>).

## 2.5 Kernelized correlation filter tracker (KCF)

In [34], Henriques et al. used a kernelized correlation filter that operates on HOG features. The key idea is to use all the cyclic shift versions of the target patch for training the classifier, instead of using dense sliding windows. Each training sample is assigned with a score generated by a Gaussian function depending on the shift amount. Using the advantages of circulant structure, the classifier is trained in Fourier domain efficiently (source code available<sup>4</sup>).

## 2.6 L1 tracker using accelerated proximal gradient approach (L1APG)

Bao et al. employed the idea of modeling the target by using a sparse approximation over a template set [35]. In this method, they solve an  $\ell_1$  norm related minimization for many times to achieve the sparse representation. Although this approach was used for object tracking successfully in the past, the main drawback was the demanding computational power requirement. In contrast to other  $\ell_1$  trackers, Bao uses a fast numerical solver that has a guaranteed quadratic convergence. Moreover, they claim that the tracking accuracy is also improved by including an  $\ell_2$  norm regularization on

<sup>2</sup> FCT Source Code: <http://www4.comp.polyu.edu.hk/~cslzhang/FCT/FCT.htm>.

<sup>3</sup> IVT Source Code: <http://www.cs.toronto.edu/~dross/ivt/>.

<sup>4</sup> KCF Source Code: <https://github.com/vojirt/kcf>.

the coefficients associated with the trivial templates (source code available<sup>5</sup>).

## 2.7 Multiple instance learning tracker (MILTrack)

Babenko et al. [36] utilized the multiple instance learning framework for object tracking, where image patches are bagged into positive and negative sets to discriminate the target from background. MILTrack method uses Haar-like features for representing the image patches. Target and background samples are, then, discriminated by using a boosting-based algorithm where a set of weak classifiers are combined to make a classification decision (source code available<sup>6</sup>).

## 2.8 Minimum output sum of squared errors tracker (MOSSE)

Correlation-based approaches are widely used for object tracking especially for their computational efficiency. MOSSE is an adaptive correlation-based algorithm that calculates the optimal filter for the desired Gauss-shaped convolution output [31]. The method has an update mechanism that adaptively changes the correlation filter depending on the target shape. This method has the lowest computational burden among the compared algorithms (source code available<sup>7</sup>).

## 2.9 Online discriminative feature selection tracker (ODFS)

In [37], an online discriminative feature selection approach is proposed where the classifier score is coupled with the importance of the patch samples. ODFS employs a feature selection mechanism where the features that optimize the objective function in steepest ascent direction for positive samples and steepest descent direction for negative samples are selected (source code available<sup>8</sup>).

## 2.10 Spatially regularized discriminative correlation filter tracker (SRDCF)

Discriminatively learned correlation filters (DCF) utilize a periodic assumption of the training samples to efficiently learn a classifier on all patches in the target neighborhood. The main contribution of [38] is mitigating the problems

arising from assumptions of periodicity in discriminative correlation filters by introducing a spatial regularization function that penalizes filter coefficients residing outside the target region. By selecting the spatial regularization function to have a sparse discrete Fourier spectrum, the filter is efficiently optimized directly in the Fourier domain. For the classification of the candidate patches, SRDCF employs HOG and gray-scale features giving a 42-dimensional feature vector at each 4x4 HOG cell (source code available<sup>9</sup>).

## 2.11 Sum of template and pixel-wise learners (Staple)

In order to construct a model that is robust to intensity changes and deformations, [39] combines two different image patch representations which are sensitive to complementary effects. Correlation-based algorithms have robust results on illumination change scenarios, but they are sensitive to deformations because of their dependency on the object shape. Color-based approaches, on the other hand, handle shape variations well, but their dependency on color hurts the performance on illumination changes. This tracking algorithm combines the translation results of two approaches in a weighted manner based on their reliability scores to achieve a higher accuracy (source code available<sup>10</sup>).

## 2.12 Ensemble of MOSSE trackers (TBOOST)

In [40], an ensemble-based object tracking method is proposed. This algorithm creates and updates an adaptive ensemble of simple correlation filters and generates tracking decisions by switching among the individual correlators in the ensemble depending on the target appearance in a computationally highly efficient manner.

## 2.13 Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection (LADCF)

In [41], an adaptive spatial regularizer to train low-dimensional discriminative correlation filters is utilized. By employing a temporal consistency constraint, a low-dimensional discriminative manifold space is formed. Adaptive spatial regularization and temporal consistency are combined to achieve a robust tracking result. The method also utilizes HOG, Color Names and ResNet-50 features to achieve better performance (source code available<sup>11</sup>).

<sup>5</sup> LIAPG Source Code: <https://github.com/lukacu/visual-tracking-matlab/tree/master/liapg>.

<sup>6</sup> MILTrack Source Code: <https://github.com/lukacu/mil>.

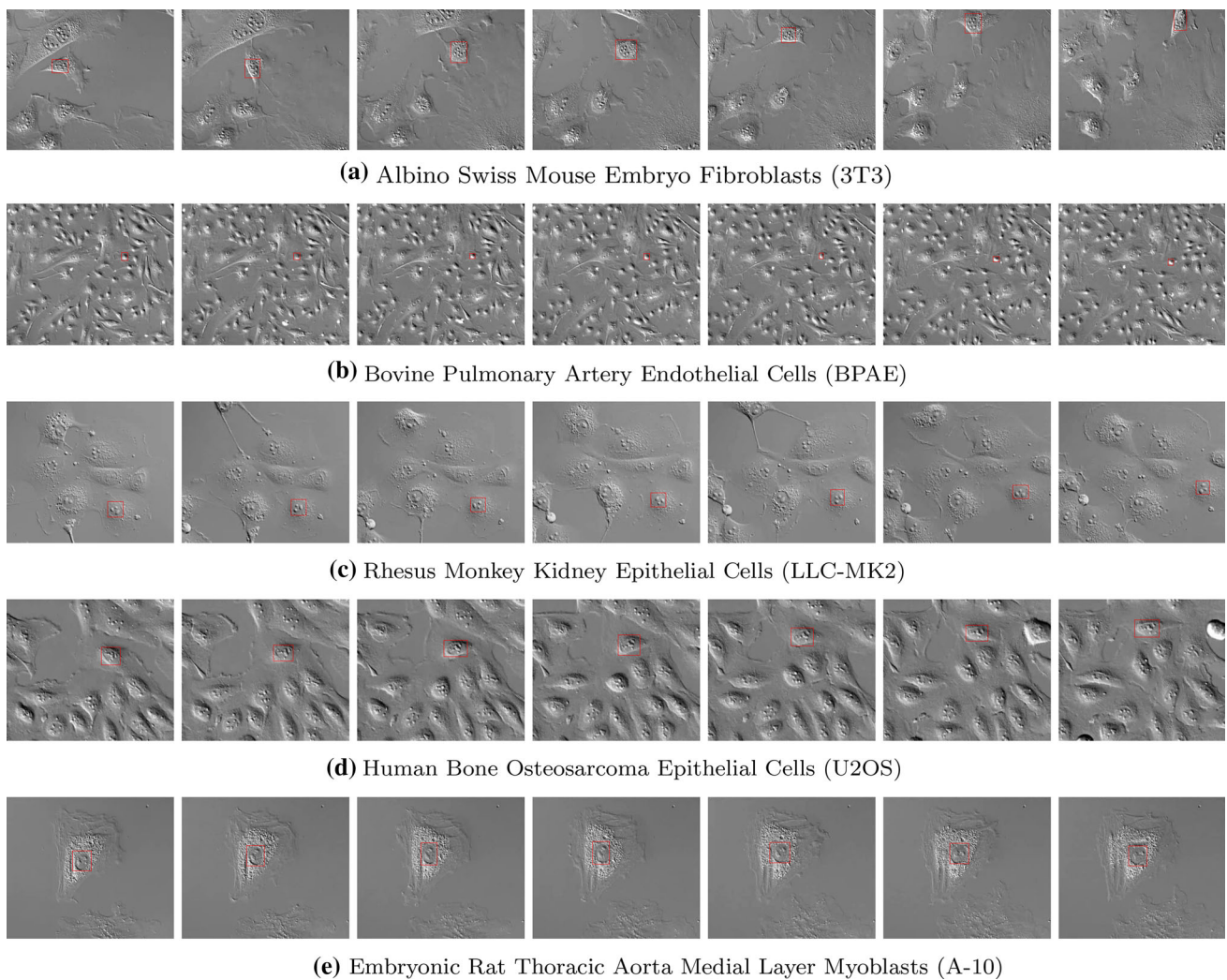
<sup>7</sup> MOSSE Source Code: <https://github.com/albertoQD/tracking-mosse>.

<sup>8</sup> ODFS Source Code: <http://www4.comp.polyu.edu.hk/~cslzhang/ODFS/ODFS.htm>.

<sup>9</sup> SRDCF Source Code: <https://www.cvl.isy.liu.se/en/research/objrec/visualtracking/regvistrack/>.

<sup>10</sup> Staple Source Code: <https://github.com/bertinetto/staple>.

<sup>11</sup> LADCF Source Code: <http://www.votchallenge.net/vot2018/trackers.html>.



**Fig. 2** Various cell image sequences for evaluation. For each sequence, ground truth bounding box that belongs to one object is depicted for frames 1, 50, 100, 150, 200, 250 and 300. The duration between two consecutive frames is 30 s

## 2.14 Learning to track at 100 FPS with deep regression networks (GOTURN)

In [42], Held et al. proposed a method for using neural networks to track generic objects by training on labeled videos. Unlike the most of the previous attempts to utilize neural networks for tracking task, GOTURN uses a simple feed-forward network with no online training in order to run in real-time. The tracker aims to learn generic object motion in training phase to track novel objects that will appear in the testing phase (source code available<sup>12</sup>).

<sup>12</sup> GOTURN Source Code: [https://github.com/foolwood/GOTURN\\_matconvnet](https://github.com/foolwood/GOTURN_matconvnet).

## 3 Experiments

We compared the tracking algorithms using the metrics described in the following subsection.

### 3.1 Performance metrics

In all the following experiments, we use two evaluation metrics, i.e., success and precision rates, used in [10].

The first metric is the success rate which indicates the percentage of frames, in which the overlap ratio between the ground truth and the tracking result is sufficiently high with respect to an appropriate threshold. A success rate plot can be generated by varying the overlap threshold between 0 and 1. In order to rank the tracking algorithms based on their success rates, we use the Area Under Curve (AUC) and Track Maintenance (TM) scores, which are derived from success

**Table 1** Success and precision rate comparison for cell motility dataset

	Success AUC	TM	Precision LA
Staple [39]	0.67	1.000	1.000
DSST [30]	0.63	0.978	0.978
CODIFF [24]	0.52	1.000	0.972
SRDCF [38]	0.56	0.976	0.959
L1APG [35]	0.57	0.989	0.949
TBOOST [40]	0.58	0.947	0.911
COV [27]	0.48	0.977	0.898
KCF [34]	0.45	0.787	0.826
MOSSE [31]	0.46	0.856	0.823
IVT [33]	0.43	0.803	0.783
LADCF [41]	0.36	0.791	0.697
FCT [32]	0.38	0.788	0.693
ODFS [37]	0.31	0.700	0.522
GOTURN [42]	0.14	0.428	0.285
MIL [36]	0.11	0.345	0.193

plots. AUC refers to the total area under a success rate plot, and TM is the ability of a tracker to maintain a track, i.e., the percentage of frames where a nonzero overlap ratio is maintained.

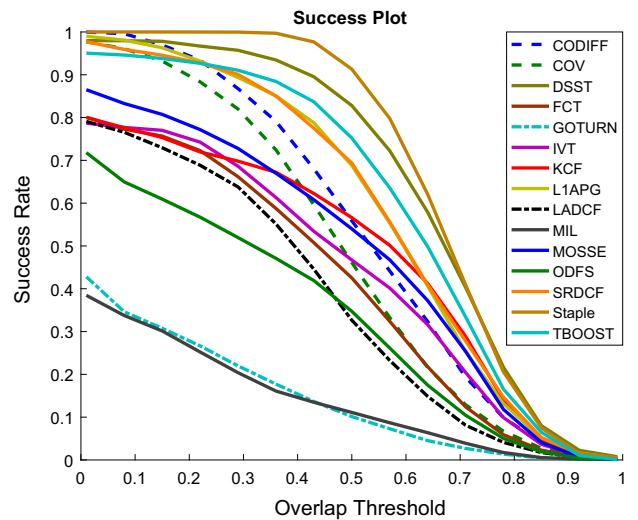
The second evaluation metric is the precision value. It denotes the percentage of the frames in which the Euclidean distance between the estimated and the actual target centers is smaller than a given threshold. The precision value demonstrates the localization accuracy (LA) of a given tracking method. In order to rank the algorithms based on their precision value, a distance threshold of 20 pixels is used in Table 1.

**3.2 Dataset**

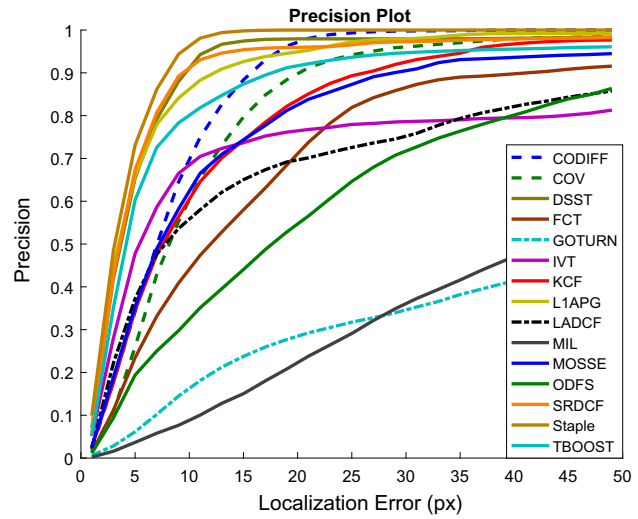
For our experiments, we used Nikon cell motility dataset [43]. In order to make a comparison between tracking algorithms, we first generated the ground truth data by annotating the cells in each frame, where every cell is considered as a new object. The dataset contains 5 different image sequences and 40 annotated objects that compose nearly 35,000 bounding box data in ground truth. Dataset contains image sequences with challenging rotation and deformation scenarios as well as different object sizes (see Figs. 1, 2).

**3.3 Results**

Overall performance results of compared visual object trackers are depicted in Fig. 3, and quantitative results are listed in Table 1. Best performing tracking algorithms for each individual data sequence are shown in Table 2.



(a) Success vs overlap threshold plot



(b) Precision vs. localization error plot

**Fig. 3** Success and precision plots for cell motility dataset

**Table 2** Best performing trackers in cell motility video sequences

Rank	3T3	BPAE	LLC-MK2	U2OS	A-10
<b>Success rate</b>					
#1	Staple	Staple	KCF	DSST	TBOOST
#2	DSST	L1APG	Staple	Staple	KCF
#3	CODIFF	DSST	DSST	KCF	CODIFF
<b>Precision</b>					
#1	DSST	Staple	Staple	DSST	Staple
#2	Staple	L1APG	SRDCF	Staple	DSST
#3	CODIFF	DSST	KCF	L1APG	KCF

Cell motility results show that Staple, DSST and CODIFF algorithms have a better precision behavior than other algorithms with a localization accuracy higher than 97 percent.

When the track maintenance scores are examined, best performing tracking algorithms are Staple, CODIFF and LIAPG. AUC scores show that Staple, DSST and TBOOST algorithms have the most successful results in terms of average success rate.

The results show that the neural network-based trackers have not been performed very well in microscopy videos, although they achieve successful results in color images. This might be caused by the fact that the deep visual features utilized by the trackers are obtained by training on the color videos captured by regular cameras. These features might not be very descriptive in microscopy videos. Therefore, training the models on microscopy datasets might increase the tracking performance.

## 4 Conclusion

In this study, we compared various state-of-the-art object tracking algorithms on a cell motility dataset and presented a framework for evaluating the performance of new cell tracking algorithms. Our experiments showed that Staple tracker which utilizes a mixture of correlation-based and color-based approaches has the best results in terms of localization accuracy, track maintenance and success rate. In general, DSST and CODIFF are other best performing methods based on the metrics used in the comparison.

## References

- Jacob, R.J., Karn, K.S.: Eye tracking in human–computer interaction and usability research: ready to deliver the promises. *Mind* **2**(3), 4 (2003)
- Majaranta, P., Bulling, A.: Eye tracking and eye-based human–computer interaction. In: Gilleade, K. (ed.) *Advances in Physiological Computing*, pp. 39–65. Springer, London (2014)
- Sotelo, M.A., Rodriguez, F.J., Magdalena, L., Bergasa, L.M., Boquete, L.: A color vision-based lane tracking system for autonomous driving on unmarked roads. *Auton. Robots* **16**(1), 95–116 (2004)
- Petrovskaya, A., Sebastian, T.: Model based vehicle detection and tracking for autonomous urban driving. *Auton. Robots* **26**(2–3), 123–139 (2009)
- Habiboglu, Y.H., Gunay, O., Cetin, A.E.: Real-time wildfire detection using correlation descriptors. In: 19th European Signal Processing Conference, 2011, pp. 894–898. IEEE
- Cao, C., Li, C., Sun Y.: Motion tracking in medical images. In: *Biomedical Image Understanding*, Chap. 7, pp. 229–274. Wiley (2015). <https://doi.org/10.1002/9781118715321.ch7>
- Li, D., Winfield, D., Parkhurst, D.J.: Starburst: a hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops. CVPR Workshops, pp. 79–79. IEEE (2005)
- Mehrubeoglu, M., Pham, L.M., Le, H.T., Muddu, R., Ryu, D.: Real-time eye tracking using a smart camera. In: 2011 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Oct 2011, pp. 1–7
- Maška, M., Ulman, V., Svoboda, D., Matula, P., Matula, P., Ederra, C., Urbiola, A., España, T., Venkatesan, S., Balak, D.M.W., et al.: A benchmark for comparison of cell tracking algorithms. *Bioinformatics* **30**(11), 1609–1617 (2014)
- Wu, Y., Lim, J., Yang, M.-H.: Online object tracking: a benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2013, pp. 2411–2418
- Li, K., Miller, E.D., Chen, M., Kanade, T., Weiss, L.E., Campbell, P.G.: Cell population tracking and lineage construction with spatiotemporal context. *Med. Image Anal.* **12**(5), 546–566 (2008)
- Friedl, P., Gilmour, D.: Collective cell migration in morphogenesis, regeneration and cancer. *Nat. Rev. Mol. Cell Biol.* **10**(7), 445–457 (2009)
- Friedl, P., Alexander, S.: Cancer invasion and the microenvironment: plasticity and reciprocity. *Cell* **147**(5), 992–1009 (2011)
- Jiang, R.M., Crookes, D., Luo, N., Davidson, M.W.: Live-cell tracking using sift features in dic microscopic videos. *IEEE Trans. Biomed. Eng.* **57**(9), 2219–2228 (2010)
- Gerlich, D., Mattes, J., Eils, R.: Quantitative motion analysis and visualization of cellular structures. *Methods* **29**(1), 3–13 (2003)
- Debeir, O., Van Ham, P., Kiss, R., Decaestecker, C.: Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes. *IEEE Trans. Med. Imaging* **24**(6), 697–711 (2005)
- Dunn, G.A., Jones, G.E.: Cell motility under the microscope: Vorsprung durch technik. *Nat. Rev. Mol. Cell Biol.* **5**(8), 667 (2004)
- Ray, N., Acton, S.T.: Motion gradient vector flow: an external force for tracking rolling leukocytes with shape and size constrained active contours. *IEEE Trans. Med. Imaging* **23**(12), 1466–1478 (2004)
- Sato, Y., Chen, J., Zoroofi, R.A., Harada, N., Tamura, S., Shiga, T.: Automatic extraction and measurement of leukocyte motion in microvessels using spatiotemporal image analysis. *IEEE Trans. Biomed. Eng.* **44**(4), 225–236 (1997)
- Hand, A.J., Sun, T., Barber, D.C., Hose, D.R., MacNeil, S.: Automated tracking of migrating cells in phase-contrast video microscopy sequences using image registration. *J. Microsc.* **234**(1), 62–79 (2009)
- Meijering, E., Dzyubachyk, O., Smal, I., et al.: 9 Methods for cell and particle tracking. *Methods Enzymol* **504**(9), 183–200 (2012)
- Chenouard, N., Smal, I., De Chaumont, F., Maška, M., Sbalzarini, I.F., Gong, Y., Cardinale, J., Carthel, C., Coraluppi, S., Winter, M., et al.: Objective comparison of particle tracking methods. *Nat. Methods* **11**(3), 281–289 (2014)
- Piccinini, F., Kiss, A., Horvath, P.: Celltracker (not only) for dummies. *Bioinformatics* **32**(6), 955–957 (2016)
- Demir, H.S., Cetin, A.E.: Co-difference based object tracking algorithm for infrared videos. In: IEEE International Conference on Image Processing (ICIP), Sept 2016, pp. 434–438
- Porikli, F., Kocak, T.: Robust license plate detection using covariance descriptor in a neural network framework. In: IEEE International Conference on Video and Signal Based Surveillance, 2006. AVSS '06, Nov 2006, pp. 107–107
- Faraki, M., Harandi, M.T., Porikli, F.: Approximate infinite-dimensional region covariance descriptors for image classification. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Apr 2015, pp. 1364–1368
- Porikli, F., Tuzel, O., Meer, P.: Covariance tracking using model update based on lie algebra. In: IEEE Computer Society Conference

- on Computer Vision and Pattern Recognition, June 2006, vol. 1, pp. 728–735
28. Tuna, H., Onaran, I., Cetin, A.E.: Image description using a multiplier-less operator. *IEEE Signal Process. Lett.* **16**(9), 751–753 (2009)
  29. Suhre, A., Keskin, F., Ersahin, T., Cetin-Atalay, R., Ansari, R., Cetin, A.E.: A multiplication-free framework for signal processing and applications in biomedical image analysis. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013, pp. 1123–1127
  30. Danelljan, M., Häger, G., Khan, F., Felsberg, M.: Accurate scale estimation for robust visual tracking. *British Machine Vision Conference*, Nottingham, 1–5 Sept 2014. BMVA Press
  31. Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M.: Visual object tracking using adaptive correlation filters. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010, pp. 2544–2550
  32. Zhang, K., Zhang, L., Yang, M.-H.: Fast compressive tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(10), 2002–2015 (2014)
  33. Ross, D.A., Lim, J., Lin, R.-S., Yang, M.-H.: Incremental learning for robust visual tracking. *Int. J. Comput. Vis.* **77**(1), 125–141 (2007)
  34. Henriques, J.F., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**, 583–596 (2015)
  35. Bao, C., Wu, Y., Ling, H., Ji, H.: Real time robust l1 tracker using accelerated proximal gradient approach. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012, pp. 1830–1837
  36. Babenko, B., Yang, M.-H., Belongie, S.: Visual tracking with online multiple instance learning. In: *IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2009*, June 2009, pp. 983–990
  37. Zhang, K., Zhang, L., Yang, M.-H.: Real-time object tracking via online discriminative feature selection. *IEEE Trans. Image Process.* **22**(12), 4664–4677 (2013)
  38. Danelljan, M., Hger, G., Khan, F.S., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: *IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 4310–4318
  39. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.S.: Staple: complementary learners for real-time tracking. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016
  40. Gundogdu, E., Ozkan, H., Demir, H.S., Ergezer, H., Akagunduz, E., Pakin, S.K.: Comparison of infrared and visible imagery for object tracking: toward trackers with superior ir performance. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2015, pp. 1–9
  41. Xu, T., Feng, Z.-H., Wu, X.-J., Kittler, J.: Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual tracking. *arXiv preprint arXiv:1807.11348* (2018)
  42. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 fps with deep regression networks. In: *European Conference Computer Vision (ECCV)*, 2016
  43. Nikon Instruments: Cell motility. <https://www.microscopyu.com/galleries/cell-motility>, 2016, Nikon’s MicroscopyU. Accessed 02 June 2017

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.