Efficient Target Recovery Using STAGE for Mean-shift Tracking

Frederick Tung John S. Zelek David Clausi
University of Waterloo
Vision and Image Processing Lab
Waterloo, Ontario, Canada
{ftung, jzelek, dclausi}@uwaterloo.ca

Abstract

Robust visual tracking is a challenging problem, especially when a target undergoes complete occlusion or leaves and later re-enters the camera view. The mean-shift tracker is an efficient appearance-based tracking algorithm that has become very popular in recent years. Many researchers have developed extensions to the algorithm that improve the appearance model used in target localization. We approach the problem from a slightly different angle and seek to improve the robustness of the mean-shift tracker by integrating an efficient failure recovery mechanism. The proposed method uses a novel application of the STAGE *algorithm to efficiently recover a target in the event of tracking failure. The* STAGE *algorithm boosts the performance of a local search algorithm by iteratively learning an evaluation function to predict good states for initiating searches.* STAGE *can be viewed as a random-restart algorithm that chooses promising restart states based on the shape of the state space, as estimated using the search trajectories from previous iterations. In the proposed method, an adapted version of* STAGE *is applied to the mean-shift target localization algorithm (Bhattacharyya coefficient maximization using the mean-shift procedure) to efficiently recover the lost target. Experiments indicate that the proposed method is viable as a technique for recovering from failure caused by complete occlusion or departure from the camera view.*

1. Introduction

The task of visual tracking involves estimating the location and/or appearance of a target from one video frame to the next. In recent years, there has been much interest in developing effective tracking algorithms for applications in areas including intelligent surveillance, traffic monitoring, vehicle navigation, and human-computer interaction [9]. The mean-shift tracker proposed by Comaniciu *et al.* is a popular tracking algorithm that has been demonstrated to be robust to small camera motion, partial occlusions, clutter, and scale changes [3]. In the mean-shift tracking algorithm, the target is represented by a colour histogram that is weighted according to an isotropic kernel. Given its estimated location in the previous frame, the target is efficiently localized in the current frame using the iterative mean-shift procedure.

Since its introduction, many researchers have proposed enhancements to the basic mean-shift tracking algorithm. Haritaoglu and Flickner developed an extended mean-shift tracker for tracking shopping groups in stores [4]. The authors applied a temporal background subtraction and motion detection method to segment shoppers, and incorporated both colour and edge information in the target appearance model. Liu *et al.* similarly assumed a static camera and proposed a method for integrating both colour and motion cues [6]. They also presented a technique for adaptively tuning the weights of the cues based on their reliability in the previous frame. Li put forward an adaptive binning colour model to improve the target appearance model [5]. Since a target's colour is often compactly distributed, the usual uniform partitioning of the colour space for building a colour histogram creates many zero-valued bins and is suboptimal. To solve this problem, Li's method first performs clustering on the object colour space and uses the resulting clusters to partition the space; thus, each histogram bin corresponds to a cluster. Each cluster is further represented using a histogram based on independent component analysis. Yilmaz replaced the radially symmetric kernel used in the basic mean-shift tracker with an asymmetric, nonparametric kernel based on the level set representation of the target contour [8]. The proposed kernel is a normalized version of the usual level set function used in contour representation, bounded by the zero level set. Yilmaz also introduced a method for automatically adapting the target scale and orientation by including these parameters as additional dimensions in the state space.

The enhancements described above generally increase the performance of the mean-shift tracker by improving the

978-0-7695-3651-4/09 $25.00 © 2009 IEEE
DOI 10.1109/CRV.2009.23

appearance model used in target localization. The enhancement proposed in this paper takes a different approach and focuses on target recovery: efficiently localizing a target in the event that the tracker loses it. Thus, the enhancement increases robustness by introducing an explicit mechanism for handling tracking failure, which may occur when a target is completely occluded over several frames or leaves and later re-enters the camera view, for example. In the proposed method, an adapted version of the STAGE algorithm [1] is applied to the mean-shift target localization algorithm to efficiently recover a lost target.

The rest of this paper is structured as follows. Section 2 describes the target localization method used in the mean-shift tracker. Section 3 covers local search algorithms and their relation to the STAGE algorithm. Section 4 explains the proposed application of STAGE for efficient target recovery in mean-shift tracking. Experimental results illustrating the operation of the proposed method are presented in Section 5. Finally, Section 6 closes with conclusions and possible directions for future work.

2. Target localization in the mean-shift tracker

In the mean-shift tracker [3], the target appearance is represented by a colour histogram that is weighted according to an isotropic kernel. Pixels closer to the centre of the kernel are assigned greater weight than those near the boundary. The tracker compares two histograms using a metric based on the Bhattacharyya coefficient. Given two m-bin normalized histograms $\hat{\mathbf{p}} = \{\hat{p}\}_{u=1\ldots m}$ and $\hat{\mathbf{q}} = \{\hat{q}\}_{u=1\ldots m}$, the distance between them is defined as [3]

$$d = \sqrt{1 - \rho[\hat{\mathbf{p}}, \hat{\mathbf{q}}]} \tag{1}$$

where

$$\rho[\hat{\mathbf{p}}, \hat{\mathbf{q}}] = \sum_u \sqrt{\hat{p}_u \hat{q}_u} \tag{2}$$

is the sample estimate of the Bhattacharyya coefficient and can be interpreted as a similarity measure.

A target is localized by finding the region that minimizes the distance (1) to the target model histogram, or equivalently maximizes the similarity metric (2). In addition to reducing the influence of peripheral background features, use of the kernel also produces a smooth similarity function, which makes it possible to apply a gradient optimization method to find the most similar local candidate [3]. Gradient information is provided by the mean shift vector, which always points in the direction of maximum increase in the density [2]. Target localization using the mean-shift procedure is much more efficient than optimized exhaustive search: Comaniciu *et al.* estimated that the computational time is approximately 24 times less [3].

3. Local search and STAGE

In general, local search algorithms try to efficiently find good approximate solutions to large-scale optimization problems by starting at a particular state, corresponding to a candidate solution, and iteratively moving to a neighbouring state until an optimum is reached. The quality of a particular state or solution is measured using an objective function, which the local search algorithm seeks to minimize or maximize. The most basic local search algorithm is the hill-climbing algorithm, also known as greedy local search, in which the neighbour with the highest objective value is chosen at each iteration [7]. Though simple, the basic hill-climbing algorithm is susceptible to becoming trapped in local optima. Stochastic hill-climbing algorithms address this problem by choosing the neighbour randomly, where a neighbour's probability of being chosen may depend on the degree of improvement in the objective function offered by the neighbour. Simulated annealing is a popular version of stochastic hill-climbing that permits some downhill moves, based on both the elapsed time and the change in objective value. Another solution is random-restart hill-climbing, which runs several hill-climbing searches from random initial states and returns the best solution found [7].

The STAGE algorithm, proposed by Boyan and Moore [1], boosts the performance of a local search algorithm by iteratively learning an evaluation function to predict good states for initiating searches. STAGE can be viewed as a random-restart algorithm that chooses promising restart states based on the shape of the state space, as estimated using the search trajectories from previous iterations.

STAGE progressively learns an evaluation function $V^\pi(\mathbf{x})$, defined as the "expected best Obj [objective] value seen on a trajectory that starts from state $\mathbf{x}$ and follows local search method π" [1, p. 79]. During each iteration, the local search method π is run. States on the resulting search trajectory are added to the training data, and $V^\pi(\mathbf{x})$ is re-approximated using a linear or quadratic regression model. Provided π is Markovian, not only the initial state but all intermediate states on the trajectory contribute to the training data since the search result would be the same had the search started at any of the intermediate states. Stochastic hill-climbing is then performed on $V^\pi(\mathbf{x})$, starting from the solution found by the local search, and the result determines the starting state of the next iteration. The Algorithm 1 box provides a summary of the STAGE algorithm; the interested reader can find more details in [1].

4. STAGE for efficient target recovery

In the context of failure recovery for the mean-shift tracker, we propose using a combination of STAGE and the usual mean-shift target localization algorithm to efficiently

Algorithm 1 Summary of the STAGE algorithm [1]

Set $\mathbf{x_0}$ to be a random starting state
while the number of states evaluated is less than a threshold **do**
  Run search algorithm π starting from $\mathbf{x_0}$; let $(\mathbf{x_0}, \mathbf{x_1}, ..., \mathbf{x_T})$ denote the resulting search trajectory and v the corresponding objective value
  Add $(\mathbf{x_i}, v)(i = 0, 1, ..., T)$ to the training set for $V^\pi(\mathbf{x})$
  Re-approximate $V^\pi(\mathbf{x})$ using linear or quadratic regression
  Run stochastic hill-climbing on $V^\pi(\mathbf{x})$ starting from $\mathbf{x_T}$; let $(\mathbf{z_0}, \mathbf{z_1}, ..., \mathbf{z_t})$ denote the resulting trajectory
  if $\mathbf{z_t} = \mathbf{x_T}$ **then**
    Set $\mathbf{x_0}$ to a new random starting state
  else
    Set $\mathbf{x_0}$ to $\mathbf{z_t}$
  end if
end while
return the best state found

Algorithm 2 Summary of the adapted STAGE algorithm for target recovery

Set $\mathbf{x_0}$ to be a random starting state
for a pre-defined number of iterations **do**
  Run Bhattacharyya coefficient maximization algorithm starting from $\mathbf{x_0}$; let $(\mathbf{x_0}, \mathbf{x_1}, ..., \mathbf{x_T})$ denote the resulting search trajectory and ρ the corresponding objective value
  Add $(\mathbf{x_i}, \rho)(i = 0, 1, ..., T)$ to the training set for $V^\pi(\mathbf{x})$
  Re-approximate $V^\pi(\mathbf{x})$ using quadratic regression
  Run stochastic hill-climbing on $V^\pi(\mathbf{x})$ starting from $\mathbf{x_T}$; let $(\mathbf{z_0}, \mathbf{z_1}, ..., \mathbf{z_t})$ denote the resulting trajectory
  if $\mathbf{z_t}$ is within the target ellipse centred at $\mathbf{x_T}$ **then**
    Set $\mathbf{x_0}$ to a new random starting state
  else
    Set $\mathbf{x_0}$ to $\mathbf{z_t}$
  end if
end for
return the best state found

recover the lost target. The local search method π is simply the Bhattacharyya coefficient maximization algorithm using the mean-shift procedure [3]. The objective function is the similarity metric ρ based on the Bhattacharyya coefficient (2). States are all the possible (x, y) pixel locations of the target. To generate a random starting state, a pixel within a pre-defined distance of the previous known target location is randomly selected. For simplicity, the L_∞ distance is used: that is, the distance between two states (x_1, y_1) and (x_2, y_2) is given by $\max(|x_1 - x_2|, |y_1 - y_2|)$. Alternatively, the starting state can be randomly selected from among all pixels in the image. This option removes the need to choose a suitable threshold distance. However, it also incurs additional computation time in terms of the number of STAGE iterations required to arrive at a good solution. In our implementation, a threshold distance of 200 pixels (approximately half the image height) is used.

We make a minor modification to the STAGE algorithm to adapt it to the particular mechanics of mean-shift tracking. Given a local maximum of the similarity metric ρ, execution of the Bhattacharyya coefficient maximization algorithm starting from a location within the target ellipse centred at the local maximum will generally converge to that same local maximum [3]. Hence, to reduce the possibility of becoming trapped in a local maximum, instead of resetting $\mathbf{x_0}$ to a new random starting state when $\mathbf{z_t} = \mathbf{x_T}$, the reset is performed when $\mathbf{z_t}$ falls within the target ellipse centred at $\mathbf{x_T}$.

Finally, the STAGE algorithm loops until the total number of states evaluated exceeds a pre-defined threshold. States on both $\mathbf{x}$ and $\mathbf{z}$ trajectories are counted towards the total. Simply to enable more intuitive comparison with the basic mean-shift tracker, we modify the loop to stop after the Bhattacharyya coefficient maximization algorithm has been run a pre-defined number of times, which we set to be ten in our implementation. During normal execution, the basic mean-shift tracker runs the Bhattacharyya coefficient maximization algorithm once per target per frame. Comaniciu *et al.* reported the successful tracking of five targets in real-time on a 1GHz PC [3].

Our proposed failure recovery algorithm using STAGE is summarized in the Algorithm 2 box.

The failure recovery algorithm is invoked when the likelihood of a target, as measured by the similarity metric ρ, falls below a threshold $l_{trigger}$. The location returned by the algorithm is accepted if its corresponding likelihood exceeds an acceptance threshold l_{accept} . In our experiments, we set $l_{trigger}$ to 0.3 and l_{accept} to 0.6.

5. Experimental Results

To illustrate the operation of the proposed target recovery algorithm, we walk through an example run using the *PencilCase* test video sequence. The test video sequences presented in this section have a resolution of 720×480 and a frame rate of 30 fps. Similar to [3], we use the RGB colour space quantized into $16 \times 16 \times 16$ bins.

Figure 1 shows the salient frames in the tracking results for the *PencilCase* sequence, in which a pencil case moves into complete occlusion behind a backpack and reappears later moving in a perpendicular direction. The traditional mean-shift tracker loses the target in frame 88. In frame

175, the target recovery algorithm is invoked (it is invoked in previous frames as well but only returns a solution satisfying the acceptance threshold in this frame). The first iteration starts with the random initial state of $(x, y) = (573, 384)$. Running the Bhattacharyya coefficient maximization algorithm yields the state $\mathbf{x_T} = (528, 379)$, with similarity $\rho = 0.1515$. The search trajectory followed by the Bhattacharyya coefficient maximization algorithm is plotted in Figure 1d. The empty training set is augmented with the states on the trajectory, each of which is associated with the objective value of 0.1515. Quadratic regression yields a flat $V^{\pi}(\mathbf{x})$, as shown in Figure 1e, so stochastic hill-climbing on $V^{\pi}(\mathbf{x})$ returns $\mathbf{z_t} = (528, 379)$ again. Hence, a new random initial state (615, 186) is generated for the next iteration. In the second iteration, the mean-shift target localization procedure starting from (615, 186) returns the same state (615, 186), with similarity $\rho = 0$. The search trajectory is plotted in Figure 1f. The single state on the trajectory, (615, 186), is associated with the objective value of 0 and added to the training set. Figure 1g shows the re-approximated $V^{\pi}(\mathbf{x})$. Stochastic hill-climbing on $V^{\pi}(\mathbf{x})$ starting from (615, 186) returns $\mathbf{z_t} = (614, 267)$, so in the third iteration, the Bhattacharyya coefficient maximization algorithm starts from (614, 267). Figure 1h shows the search trajectory plot and Figure 1i shows the approximated $V^{\pi}(\mathbf{x})$ surface after ten iterations. The algorithm returns the state (279, 252), corresponding to a similarity $\rho = 0.6666$, found in the seventh iteration.

Figure 2 shows the results for the *Wire* test sequence. In this sequence, a spool of wire is moved out of the camera view and returned at a different location in the view. Figures 2a to 2c show the frames in which the target is initialized, lost, and recovered with similarity $\rho = 0.8131$. Figures 2d and 2e show the search trajectory plot and approximated $V^{\pi}(\mathbf{x})$ surface after ten iterations.

Figure 3 shows the results for the *Tissue* test sequence. In this sequence, a tissue pack is moved behind a backpack and then the backpack is removed to eliminate the occlusion. Figures 3a to 3c show the frames in which the target is initialized, lost, and recovered with similarity $\rho = 0.6429$. Figures 3d and 3e plot the search trajectories and approximated $V^{\pi}(\mathbf{x})$ after ten iterations.

A limitation of the algorithm is that an incorrect object may be recovered if it is very similar to the target in appearance. Integration of a technique that improves the target appearance model, such as one of the mean-shift tracker extensions described in Section 1, can help mitigate this limitation by making the target appearance more distinctive.

6. Conclusion

A novel adaptation of STAGE for efficient target recovery in mean-shift tracking is presented. Target recovery using STAGE may be particularly viable as a technique for handling failure caused by complete occlusion or departure from the camera view for several frames. Like the traditional mean-shift tracker, the proposed method does not intrinsically require a static camera. As a result, the method is well-suited for scenarios in which the camera is not static and it is infeasible to rely simply on motion cues, such as when following a target using a pan-tilt-zoom surveillance camera.

A useful extension for future investigation would be the adaptive tuning of the algorithm parameters. Adjusting the number of STAGE iterations based on statistics from previous runs may further increase algorithm efficiency. Finally, as previously mentioned, integration of a tracker extension that improves the target appearance model can help maximize the probability of correctly recovering the target.

Acknowledgments

This work is funded by the Natural Sciences and Engineering Research Council (NSERC) of Canada via a postgraduate scholarship, Discovery Grants, and GEOIDE (Geomatics for Informed Decisions, a Network of Centres of Excellence).

References

[1] J. Boyan and A. Moore. Learning evaluation functions to improve optimization by local search. *Journal of Machine Learning Research*, 1:77–112, 2000.

[2] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[3] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–575, 2003.

[4] I. Haritaoglu and M. Flickner. Detection and tracking of shopping groups in stores. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:I431–I438, 2001.

[5] P. Li. An adaptive binning color model for mean shift tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(9):1293–1299, 2008.

[6] H. Liu, Z. Yu, and H. Zha. Robust mean shift tracking based on multi-cue integration. *IEEE International Conference on Systems, Man, and Cybernetics*, 6:5160–5166, 2006.

[7] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, second edition, 2003.

[8] A. Yilmaz. Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2007.

[9] A. Yilmaz, O. Javed, and M. Shah. Object tracking: a survey. *ACM Computing Surveys*, 38(4), 2006.

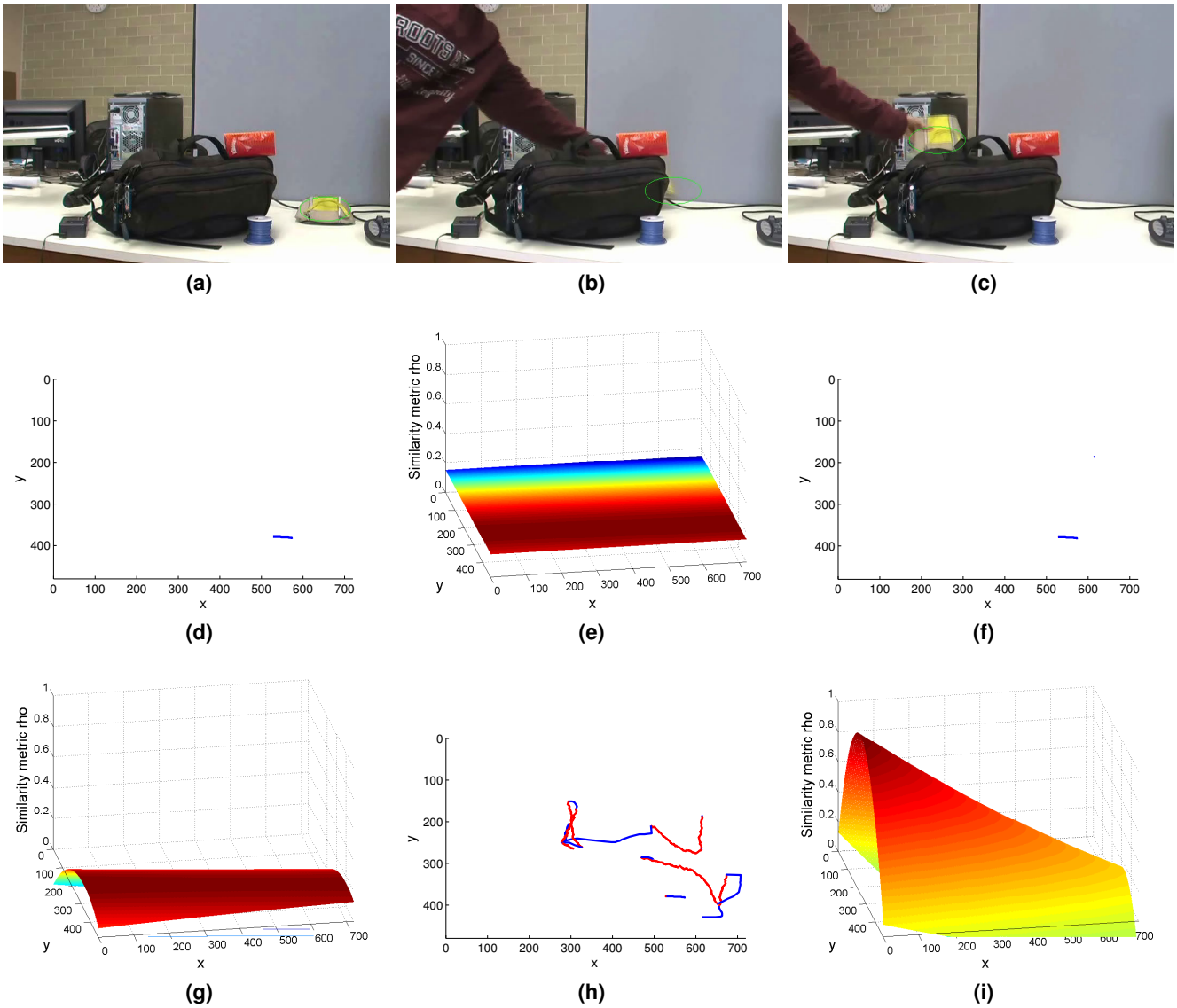
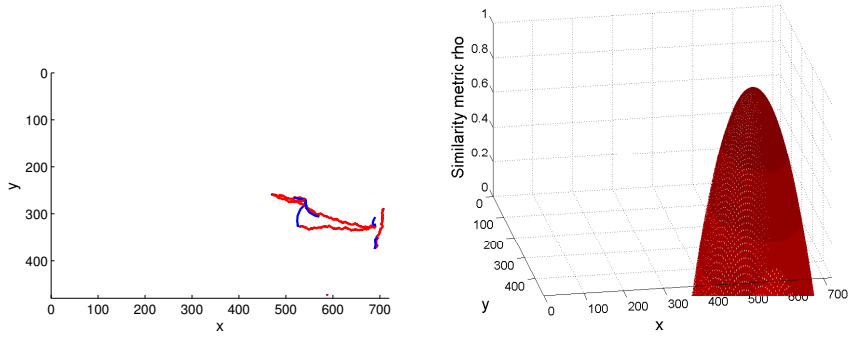


Figure 1. Operation of the target recovery algorithm on the *PencilCase* sequence. (a) Manual initialization of the mean-shift tracker in frame 1. (b) Target lost by the traditional mean-shift tracker in frame 88. (c) Target recovered by the proposed method in frame 175. Iteration 1: (d) Search trajectory after execution of Bhattacharyya coefficient maximization algorithm (line 3 in Algorithm 2 box). (e) Approximated $V^{\pi}(\mathrm{x})$ (line 5 in Algorithm 2 box). Iteration 2: (f) Search trajectories after execution of Bhattacharyya coefficient maximization algorithm, which adds only a single point in this iteration. (g) Approximated $V^{\pi}(\mathrm{x})$. Iteration 10: (h) Search trajectories after execution of Bhattacharyya coefficient maximization algorithm. Note the z trajectories from stochastic hill-climbing on $V^{\pi}(\mathrm{x})$ are shown in red. (i) Approximated $V^{\pi}(\mathrm{x})$.

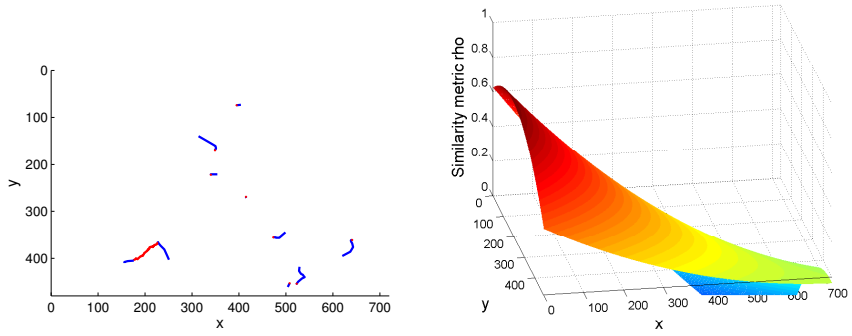
(a) (b) (c)



(d) (e)

Figure 2. Results for the *Wire* sequence. (a) Manual initialization of the mean-shift tracker in frame 1. (b) Target lost by the traditional mean-shift tracker in frame 108. (c) Target recovered by the proposed method in frame 258. (d) Search trajectories after execution of Bhattacharyya coefficient maximization algorithm (line 3 in Algorithm 2 box) in tenth iteration. Note the $\mathbf{z}$ trajectories from stochastic hill-climbing on $V^{\pi}(\mathbf{x})$ are shown in red. (i) Approximated $V^{\pi}(\mathbf{x})$ (line 5 in Algorithm 2 box) in tenth iteration.

(a) (b) (c)



(d) (e)

Figure 3. Results for the *Tissue* sequence. (a) Manual initialization of the mean-shift tracker in frame 1. (b) Target lost by the traditional mean-shift tracker in frame 69. (c) Target recovered by the proposed method in frame 207. (d) Search trajectories after execution of Bhattacharyya coefficient maximization algorithm (line 3 in Algorithm 2 box) in tenth iteration. Note the $\mathbf{z}$ trajectories from stochastic hill-climbing on $V^{\pi}(\mathbf{x})$ are shown in red. (i) Approximated $V^{\pi}(\mathbf{x})$ (line 5 in Algorithm 2 box) in tenth iteration.