Closed-Loop Adaptation for Robust Tracking

Jialue Fan, Xiaohui Shen, and Ying Wu

Northwestern University
2145 Sheridan Road, Evanston, IL 60208
{jfa699,xsh835,yingwu}@eecs.northwestern.edu

Abstract. Model updating is a critical problem in tracking. Inaccurate extraction of the foreground and background information in model adaptation would cause the model to drift and degrade the tracking performance. The most direct but yet difficult solution to the drift problem is to obtain accurate boundaries of the target. We approach such a solution by proposing a novel closed-loop model adaptation framework based on the combination of matting and tracking. In our framework, the scribbles for matting are all automatically generated, which makes matting applicable in a tracking system. Meanwhile, accurate boundaries of the target can be obtained from matting results even when the target has large deformation. An effective model is further constructed and successfully updated based on such accurate boundaries. Extensive experiments show that our closed-loop adaptation scheme largely avoids model drift and significantly outperforms other discriminative tracking models as well as video matting approaches.

1 Introduction

Object tracking is a fundamental task in computer vision. Although numerous approaches have been proposed, robust tracking remains challenging due to the complexity in the object motion and the surrounding environment. To reliably track a target in a cluttered background, an adaptive appearance model that can discriminate the target from other objects is crucial. It has been shown that in many scenarios context information can be adopted to increase the discriminative power of the model [1,2].

One way of incorporating context information is to find auxiliary objects around the target and to leverage the power of these objects to collaboratively track the target [19]. However, these methods require the presence of objects whose motion is consistently correlated to the target, which may not be satisfied sometimes. Another way is to extract the features of the background around the target, are then use them to enhance the distinction of the target against the background, either by feature selection [1], or by training classifiers [21].

One critical issue that is rarely discussed in these methods is the degradation of the model caused by the inaccuracy in the estimation of the foreground and background. Most commonly the foreground and background are divided by a bounding box or a region around the location of the target. No matter how tight

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part I, LNCS 6311, pp. 411–424, 2010.

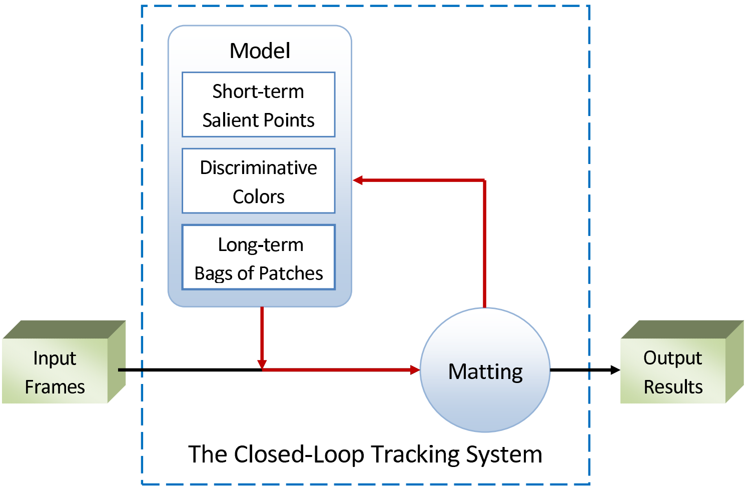

Model
Short-term Salient Points
Discriminative Colors
Long-term Bags of Patches
Input Frames
Matting
Output Results
The Closed-Loop Tracking System

Fig. 1. The framework of closed-loop adaptation for tracking

the region is, such a partition is too rough because some background regions are treated as part of the foreground, especially when the location of the target is not precise or the target is occluded. Accordingly, the updated model would gradually be degraded and thus cause drift. Grabner *et al.*[18] proposed an online semi-supervised boosting method to alleviate drift, and Babenko *et al.*[3] introduced multiple instance learning to handle the problem. Despite such efforts, an accurate boundary that clearly divides the target from the background is still desirable.

To obtain a clear boundary of the foreground, one effective way is to perform matting based on some prior information, which has been shown very successful in estimating the opacity of the foreground. The boundary can then be easily extracted from the opacity map. However, matting has never been combined with tracking before because of the gap that matting needs user interaction while tracking requires automatic processing. Video matting, although using some tracking techniques (*e.g.* optical flow) to lighten the burden of human efforts, still needs a large amount of user input and can not meet specific demands of object tracking such as automatic processing, low resolution and occlusion handling. In this paper, we bridge this gap by automatically providing suitable scribbles for matting during the tracking process and make matting work very well in the tracking scenario. Furthermore, we propose a practical model adaptation scheme based on the accurate object boundary estimated by matting, which largely avoids the drift problem. Such an interplay of matting and tracking therefore forms a **closed-loop** adaptation in an object tracking system, as shown in Fig. 1. Compared to other tracking approaches, our closed-loop tracking system has the following contributions and advantages:

1. We address the automatic scribble generation problem for matting in the tracking process. A coarse but correct partition of foreground and background is estimated during tracking, which is then used to automatically generate suitable scribbles for matting. The supply of scribbles is non-trivial. A small false scribble may lead to matting failure, while deficient scribbles could also impair the performance. In our system, the generation of scribbles is designed carefully to be correct and sufficient, which can yield comparable matting results to the methods based on user input.
2. We construct a simple but practical model for tracking, which not only captures short-term dynamics and appearances of the target, but also keeps long-term appearance variations, which allows us to accurately track the target in a long range under various situations such as large deformation, out of plane rotation and illumination change, even when the target reappears after complete occlusion.
3. Unlike other methods that tried to alleviate the aftereffects caused by inaccurate labeling of the foreground, we successfully extract the accurate boundary of the target and obtain refined tracking results based on alpha mattes. Under the guidance of such a boundary, the short-term features of the model are updated. Moreover, occlusion is inferred to determine the adaptation of the long-term model. Benefiting from the matting results, our model adaptation largely excludes the ambiguity of foreground and background, thus significantly alleviating the drift problem in tracking. Besides, object scaling and rotation can also be handled by obtaining the boundary.

2 Related Work

Object tracking has been an active research area sine early 1980s and a large number of methods were proposed during the last three decades. In the perspective of model design and update in tracking, early works tended to construct the model by describing the target itself [22,23], while recently the adoption of context information has become very popular [1,2,19,21,20,4,5,14].

The modeling of spatial context in these methods can be categorized to two levels: higher object level and lower feature level. At the higher level, the interactions between different targets are explored in multiple target tracking, either by a Markov network [5] or by modeling their social behaviors [4]. Such interactions are further extended to the auxiliary objects around the target [19]. By finding the auxiliary objects whose motion is consistently correlated to the target at a certain short period, it can successfully track the target even if the appearance of the target is difficult to discriminate. However, such auxiliary objects are not always present, which makes those methods sometimes not applicable.

At the lower level, the features of the background around the target are utilized without analyzing their semantic meanings. Feature selection can be performed by choosing the most discriminative ones between the target and its background, which is first proposed in [1]. Avidan [21] trained an ensemble of classifiers by treating the target as positive samples and the background as negative ones. These methods, however, more or less suffer from the inaccuracy in the estimation of the foreground and background, which obscures the discrimination of their model and eventually leads to drift.

The drift problem was discussed in [6], in which they proposed a partial solution for template update. Grabner *et al.*[18] proposed an online semi-supervised boosting method, and Babenko *et al.*[3] introduced multiple instance learning to avoid the drift of positive samples. However, these methods all focused on the alleviation of drift caused by foreground/background labeling errors. If an accurate boundary of the target can be obtained, such errors would be mostly reduced.

Image segmentation is one way to extract and track the object boundaries. Ren *et al.*[16] combined spatial and temporal cues in a Conditional Random Field to segment the figure from the background in each frame, and Yin *et al.*[15] modified the CRF by adding shape constraints of the target. However some tracking scenes may have cluttered backgrounds, which cause difficulties to directly extract accurate boundaries using segmentation techniques.

Compared with image segmentation, alpha matting tries to exploit the linear compositing equations in the alpha channel instead of directly handling the complexity in natural images, therefore may achieve better foreground/background separation performance based on a moderate amount of user input [7]. [8] provided an extensive review of recent matting techniques. Matting is further extended to videos by combining motion estimation approaches such as optical flow and background estimation [9,10]. But video matting can not be directly used in tracking, as they always need user interaction, which is not suitable in automated tracking methods. Moreover, they can not well handle objects with fast deformations and occlusions, which are very common in most tracking scenarios. To the best of our knowledge, our method is a first attempt to combine matting with tracking to provide shape boundary of the target and to handle occlusion.

3 Model Description

By incorporating the properties of discriminative models and descriptive models, our tracking model tries to discriminate the foreground from the background as well as maintain a long-term description for the target's appearance. Apart from the basic dynamics, the model is composed of three main components.

Short-term salient points. S_f denotes a set of salient points that are extracted from the foreground, while S_b is a set of salient points detected from the surrounding background near the target, as shown in Fig. 2(a). Currently SIFT [11] are used as salient points. Salient points are tracked in a short time period and used to generate scribbles for matting and estimate the dynamics of the model.

Discriminative colors. Color is another useful clue to discriminate the foreground from the background. We select the most discriminative colors for the foreground and the background respectively. Given a frame with known foreground and background (either by manual initialization at the first frame or

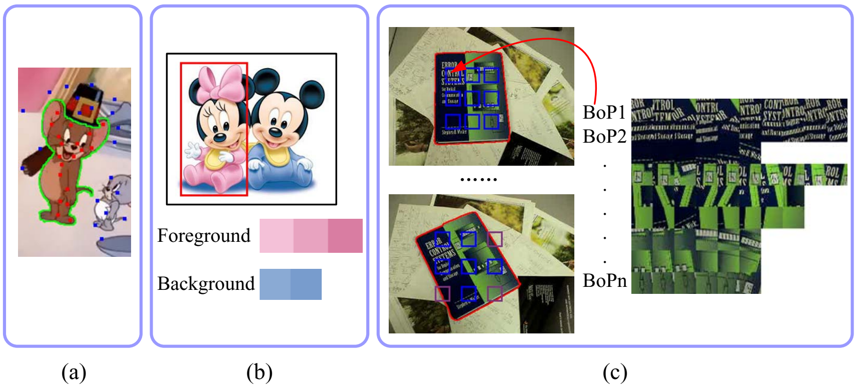


Fig. 2. Our model for tracking. (a) Short-term salient points, (b) discriminative color lists, (c) long-term bags of patches.

by refined tracking results at the following frames), we can obtain the discriminative color list of the foreground C_f based on the log-likelihood ratio of the foreground/background color histogram.[1] We can get a similar list of the background C_b, and maintain these two color lists respectively. Figure 2(b) gives us an example. In Fig. 2(b), the pink color is the most distinctive one for the foreground, while light blue is distinctive for the background. White, black and yellow exist in both the foreground and background. Therefore neither of C_f and C_b chooses them as discriminative colors. Such a description, although simple, is observed very powerful to detect the foreground and the background.

Long-term bags of patches. We also constructed a long-term model to preserve the appearance variation of the target in a long range, which helps locate the target under occlusion and deformation. Given a target region, we divide it to a $M \times N$ grid. For example, in Fig. 2(c), the grid is 3×3. At each crosspoint of the grid, a **patch** is cropped and recorded.[2] Therefore we have many patches with different time stamps at each crosspoint, which captures the variation in the local appearance at a relatively fixed position of the foreground. We call the set of all the patches at the same crosspoint a **bag** of patches BoP_i. For example, in Fig. 2(c), BoP_1 captures the long-term appearance at the top-left corner of the target. The adaptations of those bags are performed independently by foreground matching and occlusion inference, which avoids false update due to partial occlusion. The geometric information (normalized by target size) of these bags of patches is also implicitly encoded by their relative positions.

At each frame, the short-term features in the model are used to generate foreground/background scribbles for matting, and the long-term model is utilized to locate the object when it is under severe occlusion and the short-term features

[1] We divide the color to 1024 bins in HSV space(16 bins, 8 bins and 8 bins in the H, S and V channels respectively), and then get the color histogram.

[2] The patch size is $K \times K$. In practice we choose $K = 25$.

are not reliable. Once the accurate foreground boundary is determined by matting, all the components of the model will be updated accordingly, which will be introduced in the next two sections.

4 The Closed Loop: From Coarse Tracking to Matting

Given an input frame, we first use our model to perform coarse tracking. *i.e.*, locate the target and obtain a coarse but correct partition of the foreground and background. Based on such a partition, scribbles are automatically generated for matting. The matting results heavily rely on the prior information of the foreground and background. A false labeling of foreground or background may cause a drastic erroneous matte. We carefully design the scribble generation scheme to avoid false labeling and to yield good alpha mattes.

4.1 Short-Term Coarse Tracking

From frame to frame, we use two types of features to detect the foreground and background and then generate scribbles: salient points and homogenous regions.

Salient points. Consider that S_f and S_b are the sets of salient points extracted from the foreground and its neighboring background at the previous frame f^{t-1} respectively, and S'_f and S'_b are the corresponding salient point sets at the current frame f^t. First we perform SIFT matching between S_f and S'_f. However, we can not guarantee that the salient points at f^{t-1} are still salient at f^t. Therefore, for those points in S_f that do not find their matching points in S'_f, they are tracked by calculating SSD and gradient-based search to find new locations at f^t [13]. At last all the points in S_f will have matched locations at the current frame. Small image regions that cover these new locations are then labeled as the foreground. Similarly, we track all the salient points in S_b and label the regions covering their new locations as the background, as we can see in Fig. 3(b). The sizes of scribbles depend on the positions of salient points and their matching scores. If a salient point is far from the object boundary at f^{t-1} and its matching score is relatively high, the corresponding scribble will be relatively large, otherwise it will be small to avoid false labeling.

It is worth notice that in our coarse tracking process, the tracking results of these salient points are not necessary to be very accurate. *It only requires that S_f still stay in the object and S_b remain in the background*, which is robust to some fluctuations in salient points tracking. In our experiments we found that such requirements are easily satisfied by the tracking results.

Discriminative color regions. Although the regions with salient points are labeled, there are still large uncertain regions around the object, some of which are very discriminative in color space. Therefore we choose discriminative color regions as additional scribbles. Consider that the discriminative color lists C_f and C_b have been updated at f^{t-1}. At f^t, we use these two lists to detect foreground discriminative color regions and background regions at the possible locations of

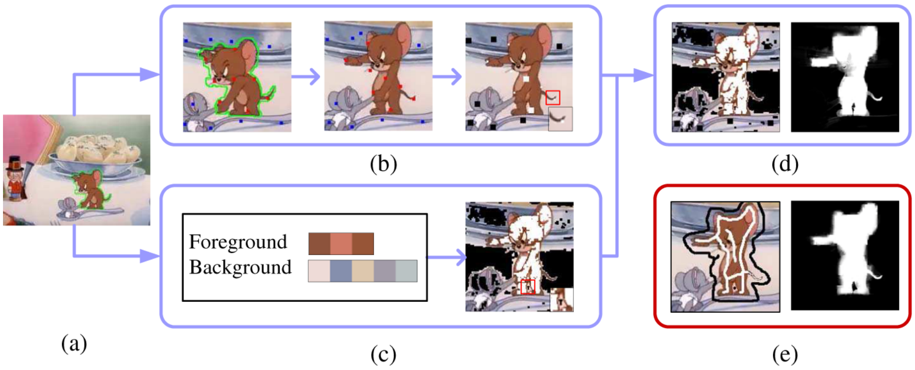


Fig. 3. Short-term coarse tracking. White regions denote foreground scribbles, while black ones denote background scribbles. (a) The boundary of the target at previous frame, (b) generating scribbles by salient point tracking, (c) Generating scribbles by the discriminative color lists, (d) final scribbles and estimated alpha matte, (e) matting result by user input.

the target. For each pixel, if its color is the same as one color in foreground discriminative color list C_f, it will be labeled as foreground. Similarly, the pixels with the same colors as in C_b are marked as background, as shown in Fig. 3(c).

The scribbles provided by salient points and the ones provided by discriminative color regions are good complements to each other. As we can see in the red square region in Fig. 3(b) (an enlarged view is provided in the lower-right corner), salient points can be detected on the tail of Jerry. And in Fig. 3(c), the region between two legs of Jerry is marked as background, where no salient points exists. Combing two categories of scribbles, the final scribbles for matting are drawn in Fig. 3(d), which ensures to produce a satisfying matte.

Given such scribbles, standard matting methods can be adopted to estimate the matte of current frame. Here we use the closed-form solution proposed in [7]. As we can see in Fig. 3(d), Our scribbles are already good and sufficient to estimate a good matte, which is very competitive against the matting result based on user input in Fig. 3(e).

4.2 Long-Term Target Locating

In most situations, the tracking results of salient points and the identification of discriminative colors are satisfying to help generate a good alpha matte. However, in some cases, such a method is not applicable, especially when the target is severely occluded and no sufficient salient points can be provided, or when the target reappears from complete occlusion. To address those problems, we use our long-term bag-of-patches model to locate the target.

Our model matching approach is based on an underlying assumption: no matter whether the target is severely occluded or first reappears, only a small part of the target is visible. Therefore, only one or two bags in our model are in the foreground at this time. That assumption significantly simplifies our matching scheme. We sequentially use one of the bags to search the space. Each maintained patch in that bag is used to find their most matched patch in the searching space (the patch with the best SSD matching score). Among those matching scores, the highest one is recorded as the matching confidence of that bag. We identify the bag with the highest confidence as the searching results. *i.e.*, the matched patch by that bag is labeled as the foreground, and the location of the model is also determined by that patch. For example, in Fig. 8, BoP_7 has the highest matching confidence at Frame 720, the target is then located according to BoP_7.

If the target is severely occluded, we can still infer its possible location according to previous tracking results, and the target can be searched in that possible region. If the target reappears from complete occlusion, then searching in the whole space may be needed. If the search space is too large, it is quite computationally expensive. Therefore we propose a coarse-to-fine method to relocate the target. We perform search every 5 pixels and find the best one, then using gradient-based SSD matching to find the local optimum. We observed that the performance is sufficiently good in experiments by this fast method.

After locating the target, the matched patch provides a foreground scribble for matting. Discriminative color detection is further performed again to generate additional scribbles around the target. Matting thus can be successfully performed using these scribbles.

5 The Closed Loop: From Matting to Refined Tracking

In the other half loop of our system, estimated alpha mattes are first adopted to refine tracking results (*i.e.* to obtain the accurate boundary and the dynamics of the target). Each component in our model can then be sequentially updated based on the clear foreground and background.

5.1 The Boundary

The alpha matte is a continuous map of opacity α. The α values near the boundary of the target are hardly 0 or 1 but some values between them. Therefore, to remove such ambiguity and to obtain a clear shape boundary of the target, a certain α threshold α_T must be chosen to cut this map.

The shape boundary of the previous frame is used as the guide to determine α_T. For example, given the boundary at Frame 166 in Fig. 4(a) and the estimated matte at Frame 167 in Fig. 4(c), by setting different thresholds, we can obtain different shape boundaries, as shown in Fig. 4(d)-(g). We assume that although the target may have large deformation, its shape in two consecutive frames should not be too different. Therefore the one having the maximum likelihood with the previous shape boundary is chosen as the boundary at the current frame. We used the contour matching method in [12] to calculate the likelihood because of its computational efficiency. The final chosen α_T is 0.2, and the boundary of the target determined by alpha matting is shown in Fig. 4(e). Compared with

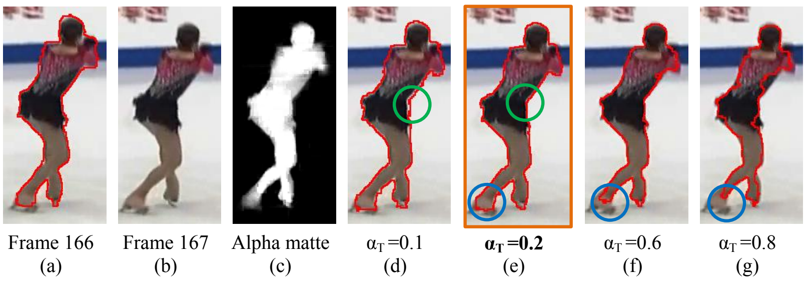


Fig. 4. Determining the boundary from estimated matte. Under the guidance of the boundary at Frame 166 in (a), the threshold is selected to be 0.2, and the boundary at Frame 167 is given in (e).

this selected threshold, a smaller α_T takes some background as foreground(the region circled using green color in Fig. 4), while a larger α_T tends to exclude some true foreground, as shown in the blue circle region in Fig. 4.

5.2 Estimating the Dynamics of the Target

The dynamics of the model are estimated from the motions of the salient points. According to the positions of the salient points at the current frame and their corresponding positions at the previous frame, their motion vectors $\mathbf{v}_i$ between these two frames are easily calculated, as shown in Fig. 5(a). Based on $\mathbf{v}_i$, a dominant motion of the entire target can be estimated. We use Parzen window method to generate a 2-D density map of salient point motion vectors.

$$f(\mathbf{v}) = \frac{1}{nh^2}\sum_{i=1}^{n} K(\frac{\mathbf{v}-\mathbf{v}_i}{h}) \tag{1}$$

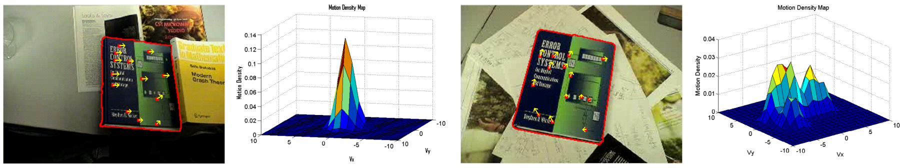
where h is the bandwidth, and $K(x)$ is the window function. Here we set $h = 3$ and $K(x)$ is a standard Gaussian function with mean zero and covariance matrix σI (I is an identity matrix). If the motion vectors of salient points present coherence, which means the entire target is also moving with a dominant motion, the motion density map must have a sharp peak(Fig. 5(b)). Let $\mathbf{v}_m$ denote the motion with the maximum density. We calculate the motion density around $\mathbf{v}_m$:

$$P(\mathbf{v}_m) = \int_{\|\mathbf{v}-\mathbf{v}_m\|<1} f(\mathbf{v})d\mathbf{v} \tag{2}$$

If $P(\mathbf{v}_m) \geq \beta$, the peak is considered very sharp, and $\mathbf{v}_m$ is the dominant motion of the entire target. The location of the target is then determined by:

$$L^t = L^{t-1} + \mathbf{v}_m \Delta t \tag{3}$$

where L^{t-1} is the location of the target at previous frame, and $\Delta t = 1$.



Dominant Motion Non-dominant Motion

(a) (b) (c) (d)

Fig. 5. Dominant motion estimation

If $P(\mathbf{v}_m) < \beta$, it indicates that the motions of salient points are not coherent, and the target may be rotating or deforming without a dominant motion, as in Fig. 5(c). Therefore, the motion estimated by salient points is not reliable. In that case, we directly use the long-term model to match the current foreground and find the location of the model, as introduced in Sect. 4.2.

5.3 Model Updating

After obtaining the clear foreground, all the components in our model are updated.

Updating salient points. Salient points are short-term features, therefore we directly re-sample new points in the foreground and the neighboring background to get obtain S_f and S_b.

Updating discriminative colors. During tracking, the background color may largely change, while the foreground may also vary due to deformation and self occlusion. Therefore, the discriminative color lists should be updated to remove invalid colors and add new discriminative colors.

Once the target is located and the boundary is estimated, we first get the color histograms of the foreground and background. Discriminative colors for the foreground and the background are then extracted respectively by calculating the log-likelihood ratio of these two histograms, as introduced in Sect. 3. For each extracted foreground discriminative color at current frame, we compare it with C_f and C_b. There are three cases:

1. C_b contains the same color, *i.e.*, one of the color features in C_b and this newly extracted discriminative color fall into the same color bin. It means that this color feature is no more discriminative for the background, and thus will be removed from C_b.
2. C_b does not contain this color while C_f does, then this color is discriminative for the foreground but already exists in C_f. No update will be performed.
3. Neither of C_b and C_f has the same color. Apparently this color feature is a new discriminative color for the foreground and will be added to C_f.

Similarly, we extract new discriminative colors in the background, and compare them with C_f and C_b. The colors in C_f which are no more discriminative are removed, and new discriminative colors for the background are added to C_b.

Updating the long-term model. A bag of patches not only contains previously appeared patches, but also records their frequency, *i.e.*, their recurrence time. The bags of patches are updated independently only when their corresponding positions (*i.e.* the crosspoints in the grid) are totally visible. By that means, only the foreground are involved in model adaptation, thus avoiding model drift caused by the intervention of background regions. Once locating the model, the positions of the bags in the model are also determined. We compare the support (a $K \times K$ square) of each bag with the foreground region. If the support of the bag is entirely inside the foreground, it is considered to be visible, and will be updated. Otherwise, it will be not updated at this time.

To update a bag of patches, we crop a new $K \times K$ patch at the bag's position, and compared it with the maintained patches by calculating their SSD. If the cropped patch is very similar to a previously maintained patch, then the frequency of the maintained patch is increased by 1, otherwise the new patch is added to the list with initial frequency as 1.[3] With such a simple but efficient model adaptation approach, the long-term local appearances of the target are effectively captured and preserved.

6 Experiments

During tracking, if the size of the target is $W \times H$, then a surrounding region with size $1.5W \times 1.5H$ is considered as its neighboring background, where salient points and discriminative color regions are extracted. We applied some morphological operators such as erosion and dilation to reduce the small noises in the scribbles. The computational cost of our approach is mostly ascribed to the matting algorithm. It is related to the amount of the pixels with un- certain alpha values before matting, which is generally dependent on the object size. In our method, much more scribbles are provided compared with user input, which makes matting faster. For example, our method can averagely process one frame per second in `Tom and Jerry` sequence without code optimization in our Matlab implementation, where the object size is around 150×150. This implies a great potential of a real-time implementation in C++. As a fast matting technique [17] has been proposed recently, the computational complexity is no longer a critical issue in our algorithm.

6.1 Comparison

We first compared our method with Collins' method [1], in which they perform feature selection to discriminate the foreground and background. In the `Tom and`

[3] Actually it is a simple clustering process, and the frequencies of the patches are the weights of the clusters.

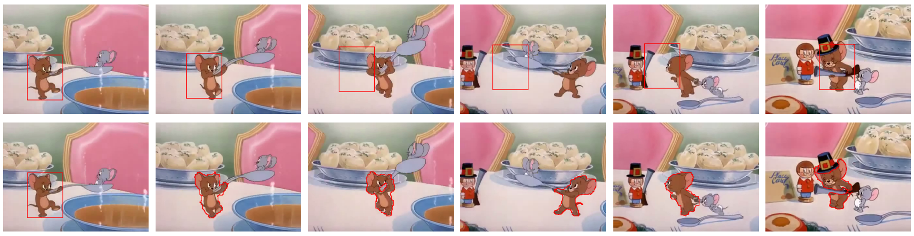

Frame 000 Frame 002 Frame 006 Frame 014 Frame 144 Frame 310

Fig. 6. Comparison with Collins' online feature selection. Top: Tracking results by online feature selection. Bottom: Tracking results by our method.

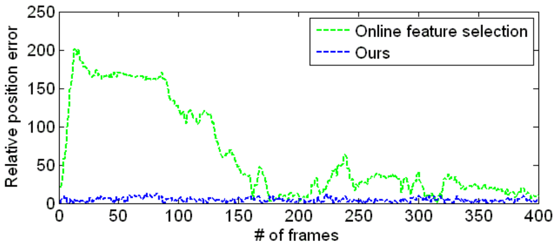


Fig. 7. Quantitative Comparison with Collins' online feature selection

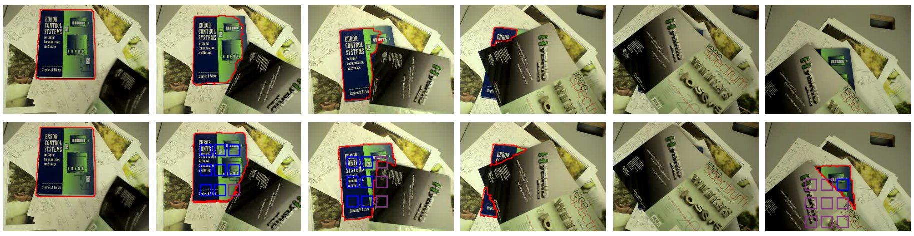
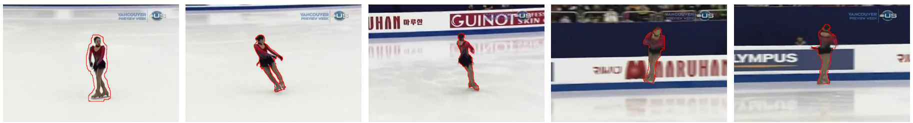

Frame 392 Frame 413 Frame 444 Frame 517 Frame 599 Frame 720

Fig. 8. Comparison with video matting. Top: Tracking results by video matting. Bottom: Tracking results by our method.

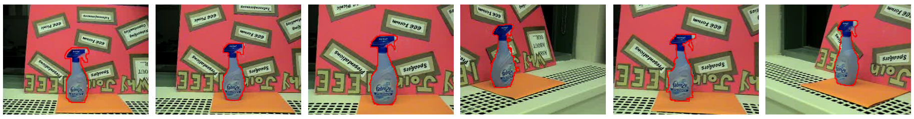
`Jerry` sequence, our approach can accurately obtain the boundary of Jerry, especially when he is holding a spoon or carrying a gun, while Collins' method drifted in the very beginning due to the fast motion, as we can see in Fig. 6. Notice that even we have a rough initialization at Frame 000 where some background is included, our method can still correctly get the boundary eventually. We also provided a quantitative comparison(Fig. 7). Our method shows a higher accuracy.

We also compared our method with video matting [9]. To make their method work in the tracking scenario (*i.e.* automatic processing), all the user input except for the first frame is removed. We both use the closed-form matting method [7] for fair comparison. As we can see in Fig. 8, in video matting the



Frame 000 Frame 015 Frame 060 Frame 114 Frame 160

Fig. 9. Tracking target with fast motion and large deformation



Frame 000 Frame 084 Frame 110 Frame 234 Frame 389 Frame 465

Fig. 10. Handling scale change and out-of-plane rotation

estimation of optical flow is not accurate at motion discontinuities and in homogeneous regions, therefore their cutout result is not satisfying. Furthermore, they cannot handle occlusion. By contrast, our method can always adaptively keep the boundary of the book. In this sequence, blue squares means that this bag is not occluded and will be updated, while purple squares means that this bag is currently under occlusion. At Frame 517, none of the bags is totally visible, the model therefore stops updating. At Frame 720, when the book reappears from complete occlusion, our model can successfully relocate it.

6.2 More Scenarios

We tested our method in more complex and challenging scenarios. The `skating` sequence has very fast motion and significant deformation. Motion blur can be clearly observed on each frame. The background keeps changing fast, especially when the skater is jumping. Given the initialization at Frame 000, Our method performs very well and gives a clear cutout for the skater, as shown in Fig. 9. Our method can also handle scaling and out-of-plane rotation in Fig. 10.

7 Conclusion

This paper introduces matting into a tracking framework and proposes a closed-loop model adaptation scheme. In our framework, the scribbles for matting are automatically generated by tracking, while matting results are used to obtain accurate boundaries of the object and to update the tracking model. Our work validates the applicability of automated matting in a tracking system, and meanwhile largely avoids the model drift problem in tracking with the aid of matting results. The proposed framework can be considered as a fundamental guideline on the combination of matting and tracking. In such a framework, each component in the closed loop can be further explored to improve the tracking performance.

Acknowledgements. This work was supported in part by National Science Foundation grant IIS-0347877, IIS-0916607, and US Army Research Laboratory and the US Army Research Office under grant ARO W911NF-08-1-0504.

References

1. Collins, R., Liu, Y., Leordeanu, M.: On-line selection of discriminative tracking features. IEEE Trans. on PAMI (2005)
2. Nguyen, H., Smeulders, A.: Robust tracking using foreground-background texture discrimination. IJCV, 277–293 (2006)
3. Babenko, B., Yang, M., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR (2009)
4. Pellegrini, S., Ess, A., Schindler, K., Van Gool, L.: You'll never walk alone: modeling social behavior for multi-target tracking. In: ICCV (2009)
5. Yu, T., Wu, Y.: Collaborative tracking of multiple targets. In: CVPR (2004)
6. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. IEEE Trans. on PAMI, 810–815 (2006)
7. Levin, A., Lischinski, D., Weiss, Y.: A closed-form solution to natural image matting. IEEE trans. on PAMI, 228–242 (2008)
8. Wang, J., Cohen, M.: Image and video matting: a survey. Foundations and Trends in Computer Graphics and Vision, 97–175 (2007)
9. Chuang, Y.Y., Agarwala, A., Curless, B., Salesin, D., Szeliski, R.: Video matting of complex scenes. In: SIGGRAPH (2002)
10. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video snapCut: robust video object cutout using localized classifiers. In: SIGGRAPH (2009)
11. Lowe, D.: Distinctive image features from scale-invariant keypoints. In: IJCV (2004)
12. Kuhl, F.P., Giardina, C.R.: Elliptic fourier features of a closed contour. Computer Graphics and Image Processing (1982)
13. Hager, G., Belhumeur, P.: Real-time tracking of image regions with changes in geometry and illumination. In: CVPR (1996)
14. Zhou, Y., Tao, H.: A background layer model for object tracking through occlusion. In: ICCV (2003)
15. Yin, Z., Collins, R.: Shape constrained figure-ground segmentation and tracking. In: CVPR (2009)
16. Ren, X., Malik, J.: Tracking as repeated figure/ground segmentation. In: CVPR (2007)
17. He, K., Sun, J., Tang, X.: Fast matting using large kernel matting laplacian matrices. In: CVPR (2010)
18. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 234–247. Springer, Heidelberg (2008)
19. Yang, M., Hua, G., Wu, Y.: Context-aware visual tracking. IEEE Trans. on PAMI, 1195–1209 (2009)
20. Wu, Y., Fan, J.: Contextual flow. In: CVPR (2009)
21. Avidan, S.: Ensemble tracking. In: CVPR (2005)
22. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: CVPR (2000)
23. Bregler, C., Malik, J.: Tracking people with twists and exponential maps. In: CVPR (1998)