Sensors & Transducers, Vol. 169, Issue 4, April 2014, pp. 25-32

Sensors & Transducers

© 2014 by IFSA Publishing, S. L.
http://www.sensorsportal.com

Scale-adaptive Local Patches for Robust Visual Object Tracking

Kang Sun, Xinwei Li
School of Electrical Engineering and Automation, Henan Polytechnic University
Jiaozuo, 454000, China
E-mail: sunkanghpu@163.com

Received: 21 January 2014 /Accepted: 7 March 2014 /Published: 30 April 2014

Abstract: This paper discusses the problem of robustly tracking objects which undergo rapid and dramatic scale changes. To remove the weakness of global appearance models, we present a novel scheme that combines object's global and local appearance features. The local feature is a set of local patches that geometrically constrain the changes in the target's appearance. In order to adapt to the object's geometric deformation, the local patches could be removed and added online. The addition of these patches is constrained by the global features such as color, texture and motion. The global visual features are updated via the stable local patches during tracking. To deal with scale changes, we adapt the scale of patches in addition to adapting the object bound box. We evaluate our method by comparing it to several state-of-the-art trackers on publicly available datasets. The experimental results on challenging sequences confirm that, by using this scale-adaptive local patches and global properties, our tracker outperforms the related trackers in many cases by having smaller failure rate as well as better accuracy. *Copyright © 2014 IFSA Publishing, S. L.*

Keywords: Scale adaptiveness, Local patch, Global feature, Visual tracking, Robustness.

1. Introduction

Given a bounding box representing the object of interest (ground truth) in a reference frame of the video sequence, the goal of tracking is to automatically determine the target's location if the target is visible in the following frames [1]. Actually, a general tracking system is an organic combination of three components [2]: 1) An appearance model, which can describe the target's properties efficiently under various circumstances. 2) A motion model, which evaluates the location of the target over time. 3) A search strategy for finding the location which having a high probability of being the target. As for the first aspect, although numerous algorithms have been proposed in recent 30 years, robust visual tracking is still a challenging mission to handle large appearance changes of target object due to background clutter, partial occlusion, illumination changes, deformation and object scale change, etc.

Most appearance models proposed to focus on robust tracking against the disturbances above can be categorized into holistic [3-5] and local [6-13] approaches. Holistic methods take advantage of global features such as color, texture, shape, motion, etc to model target's appearance. In spite of wide success, these methods often suffer drift or even failure while the target undergoing rapid structural and appearance changes, for example, partial occlusion. Online updating target template [14-16] theoretically enhance the robustness of the tracker, but at the same time increase the risk of adding invalid information that not belong to the real target, when the tracker can not locate the new position of

the target precisely. Making use of several different trackers [17, 18] seems possible to solve this problem by choosing the most repeatable results among these trackers, however its computation complexity is unbearable for practical applications.

Recently, a novel tracking methodology based on local features and local parts has been proposed. The local invariant features include SIFT [6], SURF [7], Ferns [8], ORB [9], etc. In these methods, tracking problem has been converted into wide baseline matching problem and it can void suffering from model drifting to some extent. On the other hand, the set of features is usually geometrically unconstrained, it just allows some degree of deformation by the limited spatial and structural information. Moreover, they always fail if the target regions are textureless.

For most local parts tracking, the parts in feature pool are usually loosely connected under different geometric constraints, such as global affine transformation constraint [10], local affine transformation constraint [11] and graph constraint [12]. Following these constraints, a flexible mechanism for updating appearance model is available. Kwon and Lee [13] recently proposed an approach to automatically update the topology of local pose changes. The shortcoming is that the dramatic part removal may lead to false structural changes in the geometrical model and result in failure. These problems can be handled by the scheme called two-layer model, in which each layer provides reliable constraints when updating the other layer. More recently, Cehovin et al., [14], proposed a coupled-layer visual model that combined a set of parts (local layer) together with global target appearance (global layer). It enhances the robustness while adaptively relearning targets which undergo rapid and significant appearance changes. However, in this algorithm, once a patch has been initialized, its scale keeps fixed, so it can not model target's appearance while object size changes dramatically because of camera zoom or rapid movement.

In this paper, we discuss the problem of tracking objects which undergo rapid and dramatic scale changes. To void the failure of global appearance models, we present a novel scheme that combines object's global and local appearance features. The local feature is a set of local patches that geometrically constrain the changes in the target's appearance. In order to adapt to the object's geometric deformation, the local patches could be removed and added. The addition of these patches is constrained by the global features such as color, shape and motion. The global visual features are updated via the stable local patches during tracking. To deal with scale changes, we adapt the scale of patches in addition to adapting the object bound box.

2. Our Approach

In this section we describe the main idea of the proposed algorithm. In order to make sure of efficiently updating local parts that representing the target, we employ both local and global features for appearance presentation. The local features are consisted of local patches that are organized in a style of geometrical constellation. The location of each patch in a new frame is predicted by a simple constant velocity motion model using Kalman filter. With the changes of target appearance, some patches can not match the candidate area and they should be gradually removed from the features pool. At the same time, new patches with the high possibility of being the target will be added into the features pool. The correction of this process is guaranteed by the several global features such as color and texture. In the contrast, once the new features pool is re-constructed, the local patches are used to update the global feature. The cross validation between global and local features ensures the stabilization of the appearance model updating process. To deal with large scale changes, scale adaptive patches are applied to adapting the object bound box. The working scheme of our method is shown in Fig. 1.

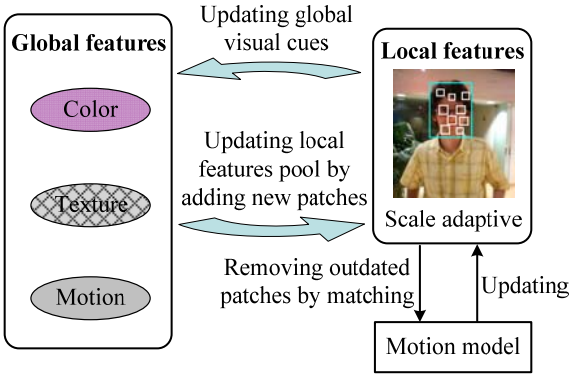


Fig. 1. The scheme of our algorithm in the processing of a single frame.

2.1. The Description of Tracking Problem

In our method, an object is represented by a set of local patch-based dynamic constellation as shown in Fig. 1. Then the appearance model $\mathcal{O}_t$ at time t is defined as:

$$\mathcal{O}_t = \frac{1}{W_t}\sum_{i=1}^{N_t} w_t^i \boldsymbol{x}_t^i , \tag{1}$$

where $\mathcal{O}_t$ denotes the center position of an object, $\boldsymbol{x}_t^i$ indicates the center position of the i[th] local patch, w_t^i represents the possibility weight that the i[th] local patch belongs to the target, $W_t = \sum_{i=1}^{N_t} w_t^i$ is a normalization factor and N_t is the number of local patches. In the following, the set of all local patches at time-step *t* is defined as $\boldsymbol{X}_t = \{\boldsymbol{x}_t^i\}_{i=1:N_t}$.

During tracking, given an initial estimate $\overline{\boldsymbol{X}}_t$ and the pixel measurements of current frame $\boldsymbol{M}_t$, the task of tracking is to search $\hat{\boldsymbol{X}}_t$ to achieve maximum value of the joint distribution of $p(\boldsymbol{M}_t, \boldsymbol{X}_t | \overline{\boldsymbol{X}}_t)$.

$$\hat{\boldsymbol{X}}_t = \arg\max_{\boldsymbol{X}_t} p\left(\boldsymbol{M}_t, \boldsymbol{X}_t \middle| \overline{\boldsymbol{X}}_t\right), \quad (2)$$

We take the advantage of affine transformation T_t which is assumed equal to all local patches, to describe the target appearance deformation during motion, and the tracking problem can be converted to find and optimize the parameters of T_t.

According to related theory of probability, $p(\boldsymbol{M}_t, \boldsymbol{X}_t | \overline{\boldsymbol{X}}_t)$ in (2) can be decomposed as follows.

$$p(\boldsymbol{M}_t, \boldsymbol{X}_t | \overline{\boldsymbol{X}}_t) = \sum_{i=1}^{N_t} p\left(\boldsymbol{r}_t^i\right) p\left(\boldsymbol{M}_t, \boldsymbol{X}_t \middle| \overline{\boldsymbol{X}}_t, \boldsymbol{r}_t^i\right), \quad (3)$$

where $\boldsymbol{r}_t^i$ is the appearance property of the i[th] patch is the local patches pool, and $p\left(\boldsymbol{r}_t^i\right)$ means the contribution of the i[th] patch representing the object region. It can be defined using the brief weight value.

$$p\left(\boldsymbol{r}_t^i\right) = w_t^i \Big/ \sum_{j=1}^{N_t} w_t^j, \quad (4)$$

The right part in (3) can be further decomposed into two parts, shown in (5).

$$p\left(\boldsymbol{M}_t, \boldsymbol{X}_t \middle| \overline{\boldsymbol{X}}_t, \boldsymbol{r}_t^i\right) = p\left(\boldsymbol{M}_t \middle| \boldsymbol{x}_t^i\right) p\left(\boldsymbol{x}_t^i \middle| \boldsymbol{s}_t^i, \overline{\boldsymbol{s}}_t^i\right), \quad (5)$$

where $\boldsymbol{s}_t^i$ and $\overline{\boldsymbol{s}}_t^i$ are respectively the new and initial locations of the i[th] patch's local neighbors with the geometrical connection of a Delaunay triangulated mesh.

$p\left(\boldsymbol{M}_t \middle| \boldsymbol{x}_t^i\right)$ in (5) measures the likelihood between the current object region and the original object template at tracking initialization, it can be defined as follows.

$$p\left(\boldsymbol{M}_t \middle| \boldsymbol{x}_t^i\right) = e^{-\lambda_v \rho\left(\boldsymbol{h}_{\text{tmp}}^i, \boldsymbol{h}_t^i\right)}, \quad (6)$$

where $\boldsymbol{h}_{\text{tmp}}^i$ and $\boldsymbol{h}_t^i$ denote the gray level histogram of i[th] patch in original object template and current object region, $\rho\left(\boldsymbol{h}_{\text{tmp}}^i, \boldsymbol{h}_t^i\right)$ is the Bhattacharryya distance between two histograms.

$p\left(\boldsymbol{x}_t^i \middle| \boldsymbol{s}_t^i, \overline{\boldsymbol{s}}_t^i\right)$ in (5) measures how far between current object region and the estimated one.

$$p\left(\boldsymbol{x}_t^i \middle| \boldsymbol{s}_t^i, \overline{\boldsymbol{s}}_t^i\right) = e^{-\lambda_g \left\| \boldsymbol{x}_t^i - T_t\left(\boldsymbol{s}_t^i, \overline{\boldsymbol{s}}_t^i\right) \overline{\boldsymbol{x}}_t^i \right\|}, \quad (1)$$

where $T_t\left(\boldsymbol{s}_t^i, \overline{\boldsymbol{s}}_t^i\right)$ represents affine transformation between $\boldsymbol{s}_t^i$ and $\overline{\boldsymbol{s}}_t^i$.

Observing (2), (5), (6) and (7), tracking problem in our method is finally an optimization problem of affine parameter T_t.

2.2. Online Updating the Local Patches Pool

We employ the histograms to represent and match the local patches in pool between template and runtime frame. These histograms features are rotation invariant and suitable for short-time tracking. However, with the deformations of appearance caused by motion, illumination changes, partial occlusion, etc, some of the patches become out of the target region and should be removed from the pool on line.

In section 2.1, we define the target appearance with patches that are associated with the brief weight w_t^i, and these weight values are not constant during tracking. After updating the patches pool, each patch at time t is estimated and its weight can be renewed as

$$w_t^i = \lambda_w w_{t-1}^i + (1 - \lambda_w) \hat{w}_t^i, \quad (8)$$

where λ_w is called forgetting factor and set as persistence constant, $\overline{w}_t^i$ is the estimated weight at time t.

The estimation of weight value $\overline{w}_t^i$ can be exactly computed by the product of two types of likelihood, respectively called 'visual consistency' $p\left(\boldsymbol{M}_t \middle| \boldsymbol{x}_t^i\right)$ defined in (6) and 'drift distance' $p\left(\boldsymbol{x}_t^i \middle| \boldsymbol{X}_t\right)$ defined as

$$p\left(\boldsymbol{x}_t^i \middle| \boldsymbol{X}_t\right) = \frac{1}{1 + e^{\lambda_D\left(mdst\left(\mathbf{x}_t^i, \mathbf{X}_t\right) - T_D\right)}}, \quad (9)$$

where $mdst\left(\boldsymbol{x}_t^i, \boldsymbol{X}_t\right)$ means the median of Euclidean distances between the positions of i[th] patch and position of every other patch in patches pool. The T_D and λ_D are constant parameters that describe the size of the target region and the influence factor of the consistency constraint respectively.

Patches with the weight value lower than a threshold W_h are labeled as either disappeared or mislocated and will be removed from the patches pool. To keep size stability of patches in pool and remove unnecessary computational burden from tracker, we also merge patches that are too close or even overlap to each other. The new patch is

initialized at the weighted average of the positions of merged patches and is given their average weight at the same time.

With the remove of outdated patches, new patches in target region should be added in patches pool to represent new appearance of target. The possibility of a new patch allocated into pool is based on its distribution defined as follows.

$$p\left(\boldsymbol{x}\middle|C_t,U_t,M_t\right)\propto p\left(C_t,U_t,M_t\middle|\boldsymbol{x}\right), \qquad (10)$$

where C_t, U_t and M_t describe the global properties of the target in aspect of color, texture and motion respectively. This means that the patches with the maximum similarity in terms of global feature will be selected as patches representing target.

Assuming that the global properties are independent at $\boldsymbol{x}$, the distribution in (11) can be decomposed as

$$p\left(\boldsymbol{x}\middle|C_t,U_t,M_t\right)\propto p\left(C_t\middle|\boldsymbol{x}\right)p\left(U_t\middle|\boldsymbol{x}\right)p\left(M_t\middle|\boldsymbol{x}\right), \quad (11)$$

By calculating a distribution map of the coming frame, we allocate the patches with the highest likelihood as the candidate ones contained in target.

2.3. Adaptive Patches for Scale Changes

Once a patch is allocated and initialized, fixed size of patch can not model the target efficiently during continuous changing of target size. For instance, when the target shrinks to sizes of similar magnitude to that of individual patches because of camera zoom or movement, much few patches can cover the whole target region, at the same time, the other patches might be forced to the nearby background region. And this will lead model drift or even mission failure. Therefore, it is necessary to change the scale of patches to adapt large scale changes of target region.

A natural idea dealing with this problem is to scale patches in proportion to the size changes of the overall bounding box. However, if the target is partially occluded, the bounding box will also shrink. Under this situation, patches should maintain their current size instead of shrinking. In addition, rapid and noisy size changes lead to instability. Allowing patch scale to rapidly respond to noise in the bounding box size engenders rapid, erroneous addition or removal of information which can irrevocably damage the adaptive appearance model. Therefore, an important issue here is how to ensure the stability while adapting the scale.

In our method, local patches are initialized with an original scale as

$$\sigma = A_o^b \big/ A_o^p , \qquad (12)$$

where A_o^p and A_o^b stand for the areas of the bounding box and a patch respectively. Scale factor σ cannot be used to change patch scale directly, because the factors of disturbance or noisy in terms of bounding box size would cause instability. Therefore, we calculate an average bounding box scale through successive *n*-frames as follows.

$$\overline{A}_k^b = \sum_{i=k-n}^{k} A_i^b , \qquad (13)$$

where A_k^b is the scale of bounding box at the frame *k*. This average scale can be expressed as $M^b = \overline{A}_k^b$.

During tracking, the average scale $\overline{A}_k^b$ is compared with the last memorized value M^b at each frame. If the scale has changed out of a safe range, then the size of patches is adjusted accordingly the following criterion.

$$A_k^p = \begin{cases} A_{k-1}^p, & \left|\overline{A}_k^b - M^b\right| < T_A \\ \lambda_A A_{k-1}^p + (1-\lambda_A)\overline{A}_k^b / \sigma, & otherwise \end{cases}, \qquad (14)$$

where T_A is the threshold of scale variation, λ_d is the persistence factor to control the speed of size adaptation, eliminating sudden, large information changes. Whenever patches are re-scaled, we also need to update the memorized bounding box

$$M^b = \lambda_A M^b + (1-\lambda_A)\overline{A}_k^b , \qquad (15)$$

Actually, T_A cannot be a fixed threshold, because the extent of any size change is relative to the current absolute size of the bounding box. Therefore, threshold T_d is itself adapted with recent box size.

$$T_A = kM^b , \qquad (16)$$

3. Experimental Results

To evaluate the performance of the proposed algorithm objectively, we here use the precision and success rate for quantitative analysis. Center location error (CLE) which is defined as the average Euclidean distance between the center locations of the tracked targets and the manually labeled ground truths is widely applied to evaluate tracking precision. However, when the tracker loses the target, the output location can be random and the average error value may not measure the tracking performance correctly. For example, if a tracker tracks an object closely for most of the video, but loses the target completely on the last several frames, the mean location error may be higher than a tracker

that sticks with the object, but not as precisely. Recently the precision plot [6] has been used to measure the overall tracking performance. It shows the percentage of all tested frames whose estimated location is within the given threshold distance of the ground truth. Typically, 20 pixels roughly corresponds to at least half of bounding box overlap between the tracker and ground truth, so we choose 20 pixels as threshold here for practically situation. However, we choose two above criterions for more general precision evaluation.

On the other hand, success rate is evaluated based on the bounding box overlap between the predicted one by the tracker and that of ground truth. Given the tracked bounding box A_t^T and the ground truth bounding box A_t^G at time-step *t*, the overlap score is defined as $\phi_t = \frac{\left|A_t^G \cap A_t^T\right|}{\left|A_t^G \cup A_t^T\right|}$, where $\cap$ and $\cup$represent the intersection and union of two regions, respectively, and $\left|\cdot\right|$denotes the number of pixels in the region. Based on the definition above, the frames with the score larger than a threshold are considered as the successfully ones. In this paper, we use the threshold value as 0.5. However, the success plot shows the ratios of successfully tracked frames at the thresholds varied from 0 to 1.

We test our tracking algorithm on several challenge and widely used video sequences (Fig. 2) with challenging factors including heavy occlusion, drastic illumination changes, pose and scale variation, non-rigid deformation, background cluster and motion blur. We compare the proposed tracker with 9 state-of-the-art methods. The parameters of the proposed algorithm are kept constant for all the experiments. For other trackers, the default parameters with the original source or binary codes are used in evaluations. The 8 trackers for comparison are: fragment tracker (Frag) [20], incremental visual tracking (IVT) method [14], online AdaBoost tracker (OAB) [15], multiple instance learning tracker (MIL) [2], tracking-learning-detection (TLD) method [16], distribution field tracker (DF) [11], compressive tracker (CT) [22] and our novel tracker. Our tracker is implemented in Matlab/C and runs at approximately 10 frames per second on an Intel i5 2.80 GHz machine with 4 GB RAM.

Fig. 2 shows the screenshots of tracking process, Fig. 3 demonstrate the performance of tracking precision for different trackers on 6 tested sequences. There are large illumination variations in sequences carDark. The appearance of the target object in this sequence changes significantly due to the cast shadows and ambient lights (see #56, #150 shown in top line of Fig. 2). Only the models of the OAB and our methods adapt to these illumination variations well. The target objects in the faceocc2 sequences are partially occluded at times (See #170, #257, #466 and #761 shown in the middle line of Fig. 2). Only the trackers of DFT, IVT and our methods can handle this situation well. The object in the tiger 2 sequence undergoes out-plane rotation (See #172, #235 shown in the bottom line of Fig. 2) which makes the tracking tasks difficult. Only DFT and the proposed algorithm are able to track the objects successfully in most frames of this sequence. And Table 1 confirms that the average center location error of our tracker is least in all the tested algorithm. The precision results at the threshold of 20 pixels for all competing trackers in different testing sequences are shown in Fig. 4 more intuitional.

Success rate with the threshold of 0.5 in Fig. 5 shows that in most tested video sequences, our enhanced MIL tracker outperforms others. This verifies the robustness of our tracker in term of quantitation.

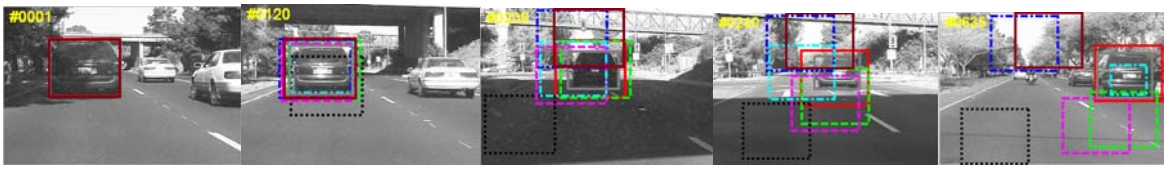
4. Conclusion

In this paper, we present a novel frame that combines object's global and local appearance features. The local feature is a set of local patches that geometrically constrain the changes in the target's appearance. In order to adapt to the object's geometric deformation, the local patches could be removed and added. The addition of these patches is constrained by the global features such as color, texture and motion. The global visual features are updated via the stable local patches during tracking. To deal with scale changes, we adapt the scale of patches in addition to adapting the object bound box. We evaluate our method by comparing it to several state-of-the-art trackers on publicly available datasets, the experimental results on challenging sequences confirm that our tracker outperforms the related trackers in many cases by having smaller failure rate as well as better accuracy.

Table 1. Average center location errors (pixels). Bond font shows best performance in all tested trackers.

Sequence	MIL	OAB	Frag	CT	TLD	IVT	DFT	Ours
carDark	43.48	2.84	36.47	119.22	27.47	8.43	58.85	**1.61**
car4	50.78	95.33	131.55	86.03	12.84	**2.15**	61.94	9.38
faceocc2	13.60	19.58	15.95	18.95	12.28	7.42	7.88	**6.70**
girl	13.67	3.70	20.67	18.85	9.79	22.46	23.98	**3.02**
sylvester	15.20	14.81	15.00	8.56	7.31	34.17	44.88	**6.38**
tiger2	27.17	251.97	113.54	28.19	37.10	105.10	**12.22**	21.87
Average CLE	27.32	64.11	55.36	46.63	17.80	29.96	34.96	**8.16**

(a) carDark



(b) car4

(c) faceocc2

(d) girl

(e) Sylvester

(f) tiger2

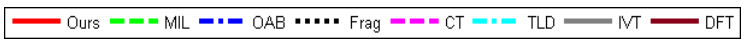


Fig. 2. Some tracking results for six sequence sets, which highlight the frames of out-of-plane rotation, occluding clutter, scale and illumination change.

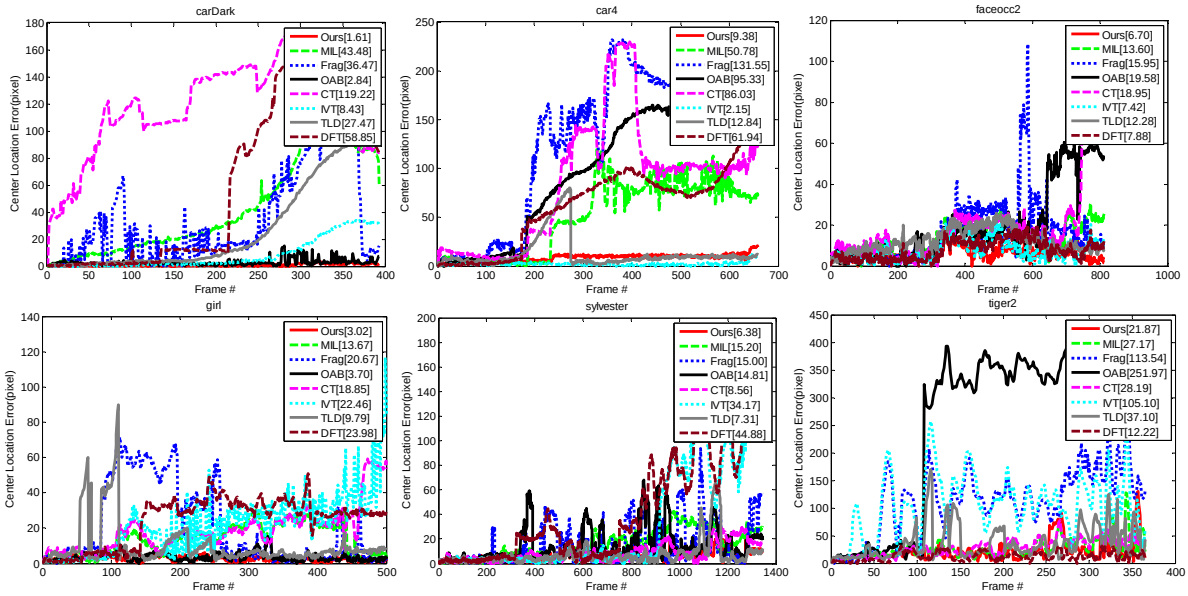


Fig. 3. Center location error plots for six video sequence.

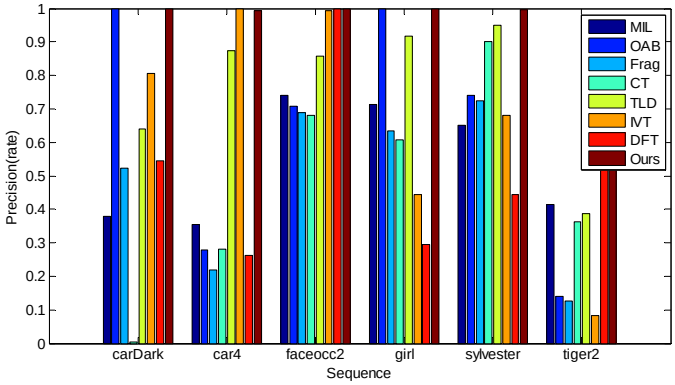


Fig.4. Precision results at a constant threshold of 20 pixels for all trackers in six testing sequences.

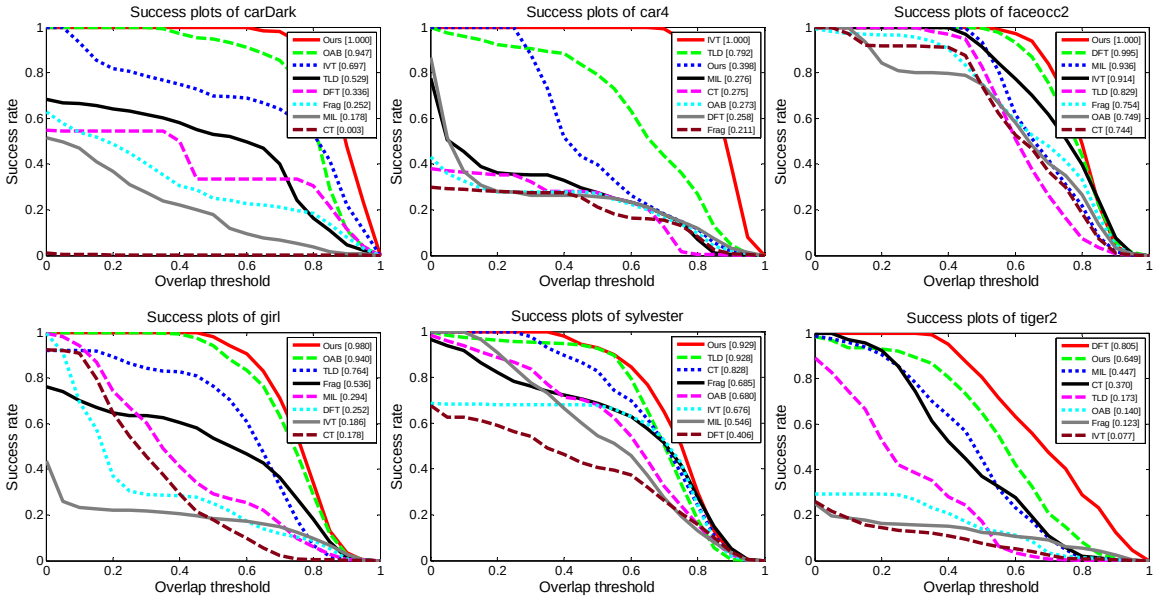


Fig. 5. Success rate plots for six video sequences.

Acknowledgements

The research work was supported by Doctor Foundation of Henan Polytechnic University under Grant No. 64998726.

References

[1]. A. Yilmaz, O. Javed, and M. Shah, Object tracking: A survey, *ACM Computing Surveys (CSUR)*, Vol. 38, No. 4, 2006, Article 13.

[2]. B. Babenko, M.-H. Yang, and S. Belongie, Robust object tracking with online multiple instance learning, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 8, 2011, pp. 1619-1632.

[3]. G. D. Hager, M. Dewan, and C. V. Stewart, Multiple kernel tracking with SSD, in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, Vol. 1, 2004, pp. 790-797.

[4]. P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, Color-Based probabilistic tracking, in *ECCV*, Vol. 1, Springer-Verlag, 2002, pp. 661-675.

[5]. S. Baker, I. Matthews, Lucas–Kanade 20 Years on: A unifying framework, *International Journal of Computer Vision*, Vol. 56, No. 3, 2004, pp. 221-255.

[6]. D. G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, Vol. 60, No. 2, 2004, pp. 91-110.

[7]. H. Bay, T. Tuytelaars, SURF: Speeded up robust features, *Computer Vision and Image Understanding*, Vol. 110, No. 3, 2008, pp. 346-359.

[8]. M. Ozuysal, M. Calonder, Fast keypoint recognition using random ferns, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, Issue 3, 2010, pp. 448-461.

[9]. E. Rublee, V. Rabaud, K. Konolige and G. Bradski, ORB: an efficient alternative to SIFT or SURF, in *Proceedings of International Conference on Computer Vision*, Barcelona, United States, 2011, pp. 2564-2571.

[10]. B. Martinez and X. Binefa, Piecewise affine kernel tracking for non-planar targets, *Pattern Recognition*, Vol. 41, No. 12, 2008, pp. 3682–3691.

[11]. V. Badrinarayanan, F. Le Clerc, L. Oisel, and P. Perez, Geometric layout based graphical model for Multi-Part object tracking, in *Proceedings of the International Workshop on Visual Surveillance*, 2008.

[12]. W. Chang, C. Chen, and Y. Hung, Tracking by parts: A Bayesian approach with component collaboration, *IEEE Transactions on Systems, Man, and Cybernetics*, Part B: Cybernetics, Vol. 39, No. 2, 2009, pp. 375–388.

[13]. J. S. Kwon and K. M. Lee, Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping Monte Carlo sampling, in *Proceedings of Computer Vision and Pattern Recognition*, 2009, pp. 1208–1215.

[14]. D. A. Ross, J. Lim, R. Lin, and M. Yang, Incremental learning for robust visual tracking, *International Journal of Computer Vision*, Vol. 77, No. 3, 2008, pp. 125–141.

[15]. H. Grabner, M. Grabner, and H. Bischof, Real-Time tracking via on-line boosting, in *Proceedings of the British Machine Vision Conference*, Vol. 1, 2006, pp. 47-56.

[16]. Z. Kalal, J. Matas, and K. Mikolajczyk, Tracking-learning-detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 7, 2012, pp. 1409-1422.

[17]. B. Stenger, T. Woodley, and R. Cipolla, Learning to track with multiple observers, in *Proceedings of the Computer Vision and Pattern Recognition*, Vol. 28, 2009, pp. 2647–2654.

[18]. J. Kwon and K. M. Lee, Visual tracking decomposition, in *Proceedings of the Computer Vision and Pattern Recognition*, Vol. 29, 2010, pp. 1269–1276.

[19]. Z. Fan, M. Yang, and Y. Wu, Multiple collaborative kernel tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, Issue 7, 2007, pp. 1268–1273.

[20]. A. Adam, E. Rivlin, and I. Shimshoni, Robust fragments-based tracking using the integral histogram, in *Proceedings of the Computer Vision and Pattern Recognition*, Vol. 1, 2006, pp. 798–805.

[21]. L. Sevilla-Lara, E. Learned-Miller, Distribution fields for tracking, in *Proceedings of the Computer Vision and Pattern Recognition*, Providence, USA, 2012, pp. 1910-1917.

[22]. K. Zhang, L. Zhang, M.-H. Yang, Real-time compressive tracking, in *Proceedings of the European Conference on Computer Vision*, Florence, Italy, 2012, pp. 758-765.

2014 Copyright ©, International Frequency Sensor Association (IFSA) Publishing, S. L. All rights reserved. (http://www.sensorsportal.com)