Li K, He FZ, Yu HP. Robust visual tracking based on convolutional features with illumination and occlusion handing. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 33(1): 223–236 Jan. 2018. DOI 10.1007/s11390-017-1764-5

Robust Visual Tracking Based on Convolutional Features with Illumination and Occlusion Handing

Kang Li^{1,2,3}, Fa-Zhi He^{1,2,*}, *Senior Member*, *CCF*, and Hai-Ping Yu^{1,2}

^{1}*State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China*

^{2}*School of Computer Science, Wuhan University, Wuhan 430072, China*

^{3}*School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China*

E-mail: likang@hubu.edu.cn; {fzhe, seaping}@whu.edu.cn

Received November 15, 2016; revised July 12, 2017.

Abstract Visual tracking is an important area in computer vision. How to deal with illumination and occlusion problems is a challenging issue. This paper presents a novel and efficient tracking algorithm to handle such problems. On one hand, a target's initial appearance always has clear contour, which is light-invariant and robust to illumination change. On the other hand, features play an important role in tracking, among which convolutional features have shown favorable performance. Therefore, we adopt convolved contour features to represent the target appearance. Generally speaking, first-order derivative edge gradient operators are efficient in detecting contours by convolving them with images. Especially, the Prewitt operator is more sensitive to horizontal and vertical edges, while the Sobel operator is more sensitive to diagonal edges. Inherently, Prewitt and Sobel are complementary with each other. Technically speaking, this paper designs two groups of Prewitt and Sobel edge detectors to extract a set of complete convolutional features, which include horizontal, vertical and diagonal edges features. In the first frame, contour features are extracted from the target to construct the initial appearance model. After the analysis of experimental image with these contour features, it can be found that the bright parts often provide more useful information to describe target characteristics. Therefore, we propose a method to compare the similarity between candidate sample and our trained model only using bright pixels, which makes our tracker able to deal with partial occlusion problem. After getting the new target, in order to adapt appearance change, we propose a corresponding online strategy to incrementally update our model. Experiments show that convolutional features extracted by well-integrated Prewitt and Sobel edge detectors can be efficient enough to learn robust appearance model. Numerous experimental results on nine challenging sequences show that our proposed approach is very effective and robust in comparison with the state-of-the-art trackers.

Keywords visual tracking, convolutional feature, gradient operator, online learning, particle filter

1 Introduction

Visual tracking has a wide range of applications such as intelligent surveillance, human interaction and virtual reality[1]. Although many tracking algorithms have been studied in recent years, it is still a challenging problem to form a robust tracker because of illumination change[2-3], occlusion[4], deformation and rotation[5]. In order to deal with such problems, researchers focus on exploiting observation model, such as boosting[6-7], structure SVM[8], sparse representation[9-13] and subspace learning[14], which can be divided into two categories: generative model and discriminative model.

In general, generative trackers typically learn an appearance model by a generative process and then search for the most similar target according to reconstruction error. IVT[14] incrementally learns a low dimensional PCA subspace representation, which can effectively model smooth pose variation. L1 tracker[9] assumes that the target could be represented by a sparse linear combination of target templates and trivial templates. After solving an l1-regularized least squares

Regular Paper

This paper is supported by the National Natural Science Foundation of China under Grant No. 61472289 and the National Key Research and Development Project of China under Grant No. 2016YFC0106305.

*Corresponding Author

problem, the candidate sample with the smallest projection error is selected as the target. Through a tracking decomposition scheme, VTD[15] shows that the target can be represented by a linear combination of object templates and trivial templates. Compared with discriminative algorithms, such generative trackers often get more accurate location when the target changes smoothly.

On the other hand, discriminative trackers usually train a classifier to separate the target from the background. For classifier training, these trackers often crop patches near the target location as positive samples and crop patches far away from the target location as negative samples. Finally, such trackers select the sample with the maximum classification score as the target. Thanks to the improvement of machine learning, several sophisticated algorithms have been applied to tracking, such as boosting, SVM and Bayesian. OAB[6] tracker uses multiple instance learning instead of traditional supervised learning and adapts the classifier while tracking the object. Moreover, it selects the most discriminating features for tracking resulting in stable tracking results. STRUCK[8] presents a framework for adaptive tracking based on structured output prediction. It uses a kernelized structured output support vector machine with a budgeting mechanism as appearance model for real-time applications. CT[16] trains appearance model based on the features which are extracted from multi-scale image feature space. It builds an online update naive Bayesian classifier to separate the target from the background, which leads to a real-time and accurate tracker. Compared with generative trackers, discriminative trackers are more robustness because purely generative trackers cannot handle complicated background well.

2 Related Work

To form a robust tracker, Wang *et al.*[1] stated that the features used in tracking systems play the most important role. A good feature can significantly improve the tracking performance even with a simple classifier.

There are many hand-crafted features used in tracking algorithms, such as local binary patterns[17], haar-like features[18-19], contour features[20-23] and other descriptors[24]. Recently, convolutional neural network (CNN)[25-27] is widely used in many image processing fields such as image classification, object recognition and visual tracking. In their convolutional layer, features are extracted by convolving with several filters, which are trained from raw data offline with little human intervention.

However, although these convolved features are robust, the filter training process is time consuming and the model trained using a large amount of general data may not suit tracking specified target. To overcome these weaknesses, we propose to use well integrated first-order derivative edge gradients instead of CNN filters to extract convolutional features.

On one hand, for object tracking, the initial targets always have clear contour features which are robust to illumination change. On the other hand, in image processing, first-order derivative edge gradients are often used to extract contour by convolving them with images. Among these first-order derivative edge gradients, Prewitt and Sobel operators are two complementary gradient operators. Related studies in facial expression detection show that the Prewitt operator is more sensitive to horizontal and vertical edges while the Sobel operator is more sensitive to diagonal edges[28-29].

Fig.1 presents a car under different light conditions (red rectangles) and corresponding contour features (green rectangles). In this figure, the contour features are gained by convolving target image with different Prewitt and Sobel operators. The yellow ovals indicate the similar structure of these contour features, which demonstrates that these features remain stable even under different light conditions. Therefore, in this paper, we integrate several Prewitt and Sobel operators to extract convolved contour features which form the appearance model of the target. The main contributions are listed below.

1) A novel appearance model based on multiple contour features is proposed. To extract contour features, we carefully select and integrate several Prewitt and Sobel operators to convolve with the target image. These contour features are robust to illumination change, which makes our algorithm able to track the target under complex light conditions.

2) We analyse and show that the bright parts of the contour features could provide more important information than dark parts. Therefore, we propose to measure the similarity between candidate samples and to only use bright pixels in our trained appearance model. Because occlusions are always smooth and have few contours, this strategy could exclude obstructed areas which help to solve partial occlusion problem.

3) After evaluating the target location at the end of each frame, a corresponding incrementally update method is presented to adapt appearance change, which shows robustness in our experiment.

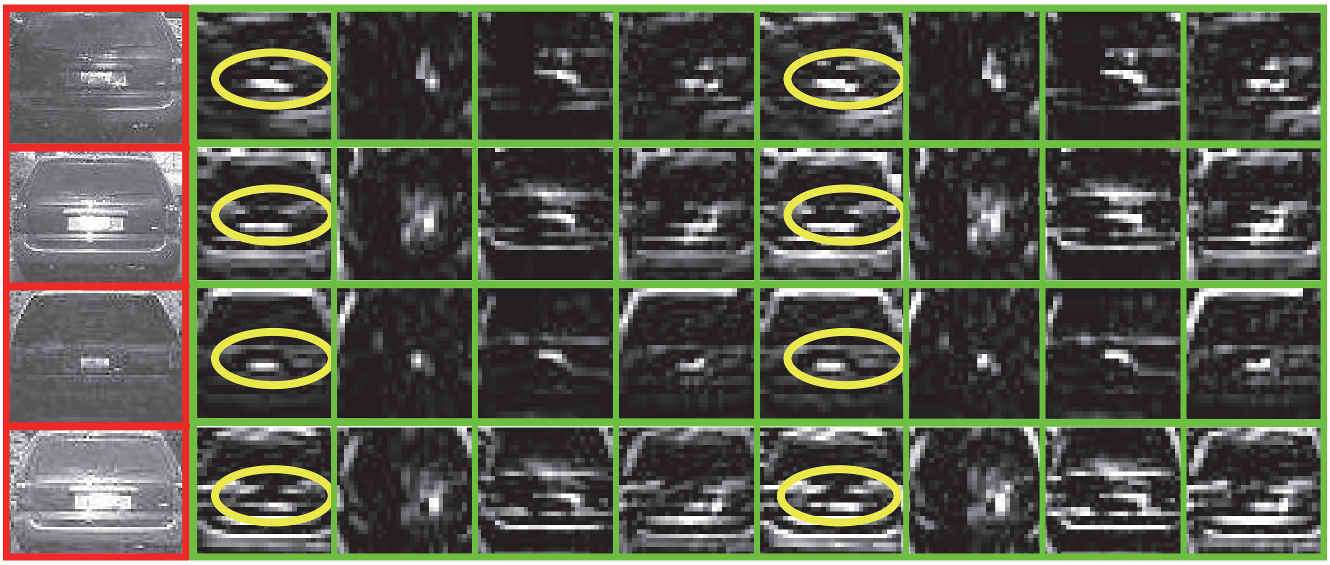
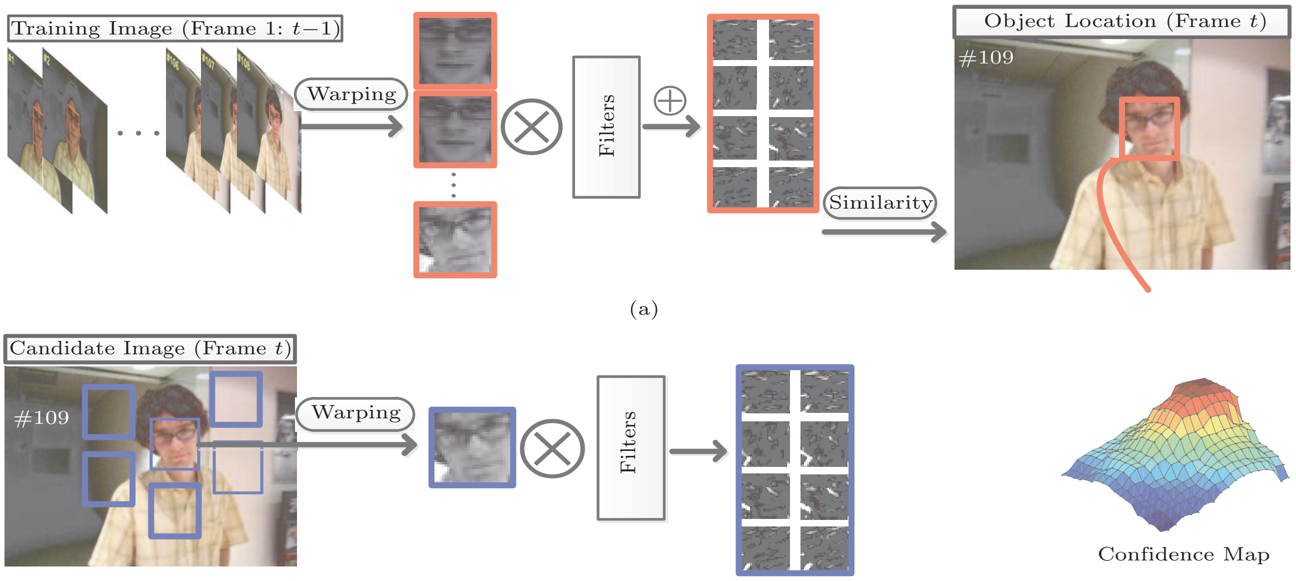


Fig.1. Example of contour feature. This is a target car (red rectangle) under four different light conditions. Although each target image has different shadow parts, their corresponding contour features (eight different contour features within the green rectangle) have similar layout (for example, the yellow ovals), which demonstrates the robustness of contour feature to the illumination change.

3 Proposed Algorithm

Fig.2 summarizes the main steps of our algorithm. In our tracker, before extracting features, every sample should be warped to the same size. At the first step, features are extracted by convolving the training images with a bank of filters. These features are warped together and form a long feature vector. Red rectangles in Fig.2(a) are the training images and their corresponding convolved contour features. Second, features are extracted and warped from candidate samples in the same way with the last step. Green rectangles in Fig.2(b) are the candidate samples and their corresponding convolved contour features. Third, the confidence map is formed by calculating the similarity between the appearance model and each candidate sample. Finally, we gain the new target sample which has the maximum



Training Image (Frame 1: $t-1$)
Warping
Filters
Similarity
Object Location (Frame t)
#109
(a)
Candidate Image (Frame t)
#109
Warping
Filters
Confidence Map
(b)

Fig.2. Main flow of our proposed algorithm. (a) Training images and their corresponding convolved contour features. (b) Candidate samples and their corresponding convolved contour features.

score in confidence map. At the end of each frame, to adapt appearance change, we incrementally update our model using the new target. Our tracker iteratively performs these steps above to track the target frame by frame.

3.1 Image Representation

In this paper, we aim to handle illumination problem. The initial appearance of a target always has a clear contour, which is light-invariant and robust to illumination change. On the other hand, features play an important role in tracking, among which convolutional features have shown favorable performance. Therefore, we adopt convolved contour features to represent the target. Generally speaking, first-order derivative edge gradient operators are efficient in detecting contours by convolving them with images. Especially, the Prewitt operator is more sensitive to horizontal and vertical edges, while the Sobel operator is more sensitive to diagonal edges[28-29]. Inherently, Prewitt and Sobel are complementary with each other. Technically speaking, we design two groups of Prewitt and Sobel edge detectors to extract a set of complete convolutional features, which include horizontal, vertical, and diagonal contour features.

In this subsection, we will introduce how to build our appearance model which contains several convolved contour features. Before extracting features, each sample x should be warped to the same size of $m \times m$ pixels (the size of 32×32 pixels is an empirical value in many papers[9,11,14] and also in our paper). Next, contour features are extracted by convolving these normalized samples with several complete Prewitt and Sobel operators. These operators are listed in (1):

$$\begin{aligned}
\boldsymbol{f}_1 &= \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{f}_2 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \\
\boldsymbol{f}_3 &= \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix}, \quad \boldsymbol{f}_4 = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \\
\boldsymbol{f}_5 &= \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, \quad \boldsymbol{f}_6 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \\
\boldsymbol{f}_7 &= \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}, \quad \boldsymbol{f}_8 = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix},
\end{aligned} \tag{1}$$

where $\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \boldsymbol{f}_4$ are Prewitt operators and $\boldsymbol{f}_5, \boldsymbol{f}_6, \boldsymbol{f}_7, \boldsymbol{f}_8$ are Sobel operators. These two-type gradient operators are complementary because the Prewitt operator is more sensitive to horizontal and vertical edges than to diagonal edges while the reverse is true for the Sobel operator. Therefore, horizontal, vertical and diagonal edge contour features could be extracted from these well integrated Prewitt and Sobel operators. In order to illustrate how to combine our contour features, we first give some notations. Given the i-th operator $\boldsymbol{f}_i$ and a patch x, the response is denoted as (2):

$$\boldsymbol{h}_i(x) = \boldsymbol{f}_i \otimes x, i = 1, ..., 8, \tag{2}$$

where $\boldsymbol{h}_i \in \mathbb{R}^{(m-w+1)\times(m-w+1)}$. Let $H(x) = (h_1(x), ..., h_8(x))$ denote features of patch x. Then $\boldsymbol{V}(x) = vec(H(x))$ is to vectorize $H(x)$, where $\boldsymbol{V}(x) \in \mathbb{R}^{(m-w+1)^2}$ is a long vector; $w = 3$ is the operator size. At the initialization step, the target template is initialized to $s = \boldsymbol{V}(x_1^*)$, which is updated incrementally to adapt appearance change, where x_1^* represents the target patch in the i-th frame.

3.2 Particle Filter

Our tracking process is driven by a particle filter framework. From the view of Bayesian estimation, the tracking problem is regarded as an inference task in Markov model with latent variables. Let $x_t = (c_x, c_y, s_w, s_h)$ denote the target state at frame t, where c_x and c_y denote center location, and s_w and s_h denote the scale at x and y orientation respectively. Given a set of observed images $I_{1:t} = \{I_1, ..., I_t\}$, we aim to estimate the state variable x_t using Bayes' theorem as (3).

$$\begin{aligned} & p(x_t|I_{1:t}) \\ & \propto p(I_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|I_{1:t-1}) \mathrm{d}x_{t-1}, \end{aligned} \tag{3}$$

where $p(x_t|x_{t-1})$ denotes the dynamical model, and $p(I_t|x_t)$ denotes the observation model which governs the tracking process. In order to develop a robust tracker for generic applications, the motion of particles between two neighbor frames is designed by Brownian motion. Each parameter in x_t is modeled independently by a Gaussian distribution around its former state. The state transition between frames is modeled as:

$$p(x_t|x_{t-1}) = N(x_t; x_{t-1}, \boldsymbol{\Omega}), \tag{4}$$

where N is the Gaussian distribution symbol, and $\boldsymbol{\Omega} = diag(\sigma_x, \sigma_y, \sigma_w, \sigma_h)$ is a diagonal matrix (its elements are the variance corresponding to the state variables).

3.3 Observation Model

In this subsection, we will discuss observation model in details. First, we explain the motivation behind our model by Fig.3. In Fig.3(b), the image patches bounded in the yellow rectangle are the template features trained using former evaluated targets; image patches bounded in the red rectangle are the target features extracted from Fig.3(a); image patches within the green rectangle are the non-target features extracted from the green area in Fig.3(a). We can see that the bright parts of the image (yellow oval) could provide more useful information than the dark parts to judge the target. Therefore, to calculate the similarity, we propose a method to compare the bright part of the contour feature image only.

At time step t, the target state is represented by a set of particles $a_t^{(i)}$. To compare the similarity between a particle $a_t^{(i)}$ and our appearance model, we first translate the particle into a gray scale and normalize it to $m \times m$.

To simplify the problem, we first define two functions.

1) Function $I_x = M(x)$ returns the index of which the corresponding elements are lower than the median value of vector $\boldsymbol{x}$.

2) Function $zero(\boldsymbol{A}, I)$ is to set $\boldsymbol{A}(I) = 0$, where $\boldsymbol{A}$ is a vector and I is an index set.

Let $z = zero\left(s, I_s\right)$ and $z_t^{(i)} = zero\left(V\left(a_t^{(i)}\right), I_s\right)$, and then the target state could be calculated by (5):

$$x_{t+1}^{*} = \arg\max_{a_{t+1}^{(i)}} O\left(z\right) \times O\left(z_t^{(i)}\right), \tag{5}$$

where $O\left(s\right)$ is the normalization function which processes by subtracting the mean and doing l_2 normalization. Fig.3 shows why our proposed method could deal with pose variation and occlusion problem. In Fig.3, the bright part of the trained template is similar to the target patch (yellow oval) and dissimilar to the non-target patch. When we compare the similarity between the training template and target patch, only the bright part is involved into calculating. At last, the occluded part, which is always smooth without contour (light blue oval), could be filtered out. Therefore, although the target is occluded, our tracker can also locate it. At the end of each frame, we set the new probability distribution function of the particle as

$$p\left(x_t | a_t^{(i)}\right) \propto \exp\left(-\left\|O\left(z\right) - O\left(z_t^{(i)}\right)\right\|_2^2\right). \tag{6}$$

Through normalizing and putting such probabilities onto the search region of the t-th frame, we get the confidence map in Fig.2.

However, object appearance often changes during tracking. In this paper, we propose an online update method to adapt appearance change. After getting the new target, we update our model by (7):

$$s \leftarrow \lambda s + \left(1 - \lambda\right) s', \tag{7}$$

where $s' = V\left(x_{t+1}^{*}\right)$, and λ is the learning rate. Higher learning rate means aggressive model update, and vice versa. The proposed tracking algorithm is summarized in Algorithm 1.

4 Experiment and Analysis

In this section, we will discuss the experimental details and analyze results. To evaluate the robustness of our tracker, we test it on several videos which are widely used in other tracking papers.[7,16,30-37] These videos are Bike, Car4, CarDark, David, David2, David3, Dog1,

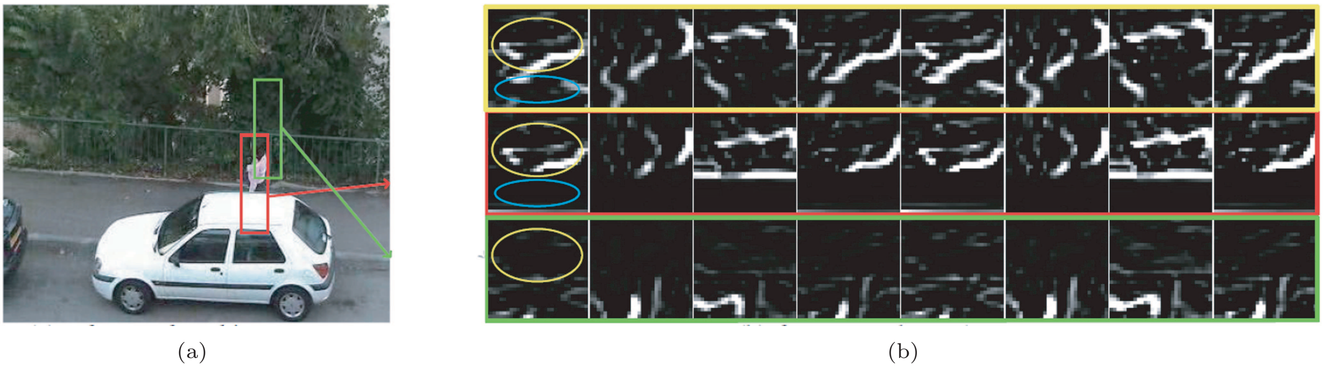

(a) (b)

Fig.3. Example of how our approach works. (a) Frame of tracking sequence. (b) Features and template. The red rectangle represents the target location in (a) and its corresponding convolutional features in (b). The green rectangle represents a non-target area in (a) and its corresponding convolutional features in (b). The first row of yellow rectangle image in (b) is the trained template.

Algorithm 1. Overview of Proposed Tracking Algorithm

```
Input:
    Initialize at time step 1: $t = 1$; $s = V(x_t^*)$; $\left\{a_t^{(i)} = x_1^*\right\}_{i=1}^{600}$; $(t+1)$-th frame data
    Other time steps: $a_t^{(i)}$; $s$; $(t+1)$-th frame data
1: Each particle $a_t^{(i)}$ moves according to (4) and get the new particles $a_{t+1}^{(i)}$ at time step $t+1$
2: Calculate new target state $x_{t+1}^*$ according to (5)
3: Set the probability distribution function of these particles according to (6) and resample them
4: Update $s$ according to (7)
5: Let $t \leftarrow t+1$
Output:
    New particles $a_{t+1}^{(i)}$ and $s$
```

FaceOcc1, Girl, Singer1, Trelis, and Woman. These videos contain several special cases in real-world context such as occlusion, blur, complex background, and pose variation. Table 1 lists the characteristics of these videos. In Table 1, complex background denotes that the background is similar to the target; background interference denotes that some background areas are contained in the bounding box. Pose variation denotes the heavily appearance change of the target. Scale change denotes that the size of the target changes during tracking. Illumination change denotes that the light condition changes in the scene. To show our tracker's overall performance, we compare our tracker with several other state-of-the-art trackers on these challenging videos. These compared trackers are CT[16], DFT[30], LOT[31], TLD[32], LSK[33], CSK[34], MIL[7], LSST[35], RPT[36] and OAFT[37].

4.1 Experimental Setup

In our experiment, the object location in the first frame is given and our model is trained in gray scale. In our experiment, our tracker is running on MATLAB (PC parameters: i5 4590 8 GB RAM 64-bit) with five frames per second without any optimization. The parameters used in the particle filter framework are listed below.

The size of normalized image m is set to 32; the learning rate λ is set to 0.9, which controls the update speed of our appearance model; particle transition parameters are set to $\sigma_x = 8$, $\sigma_y = 8$, $\sigma_w = 0.01$ and $\sigma_h = 0.01$; the number of particles is 600. Although increasing particles' number may result in a more accurate location, the speed will trend to decrease.

4.2 Evaluation Methodology

To provide the comprehensive assessment of these trackers, we use three indicators to evaluate performance.

The first method is center location error (CLE), which is defined as the Euclidean distance between the center location of tracking results and ground truths labeled by hand. The number behind each tracker is the AUC (area under the curve) value which demonstrates the average distance error. However, the CLE method is ineffective when a tracker drifts from the target.

The second method is overlap success rate plots

Table 1. Characters of Test Sequences

	Complex Background	Background Interference	Occlusion	Pose Variation	Scale Change	Illumination Change
Bike	×	√	×	√	×	×
Car4	×	×	×	×	√	√
CarDark	√	√	×	×	√	√
David	×	×	×	√	√	√
David2	√	×	×	×	×	×
David3	×	×	√	√	×	√
Dog1	×	×	×	√	√	×
FaceOcc1	×	×	√	×	×	×
Girl	×	×	√	√	√	×
Singer1	√	×	√	√	√	√
Treillis	√	×	×	√	√	√
Woman	×	×	√	√	×	×

(SR) which can evaluate the average efficiency of a tracker. Before we introduce the SR method, let us first define overlap score. An overlap score S is defined as $S = \frac{area(R \cap G)}{area(R \cup G)}$, where R denotes the tracking result region, G denotes the ground truth, $\cap$ and $\cup$ represent the intersection and union of these two regions respectively, and $area()$ is a function which returns the number of pixels of a region. If S is larger than some threshold, this frame will be labeled success; otherwise it will be labeled failure.

The third method is precision error plots (PE) based on CLE. If the CLE of a frame is lower than some thresholds, this frame will be labeled success; otherwise it will be labeled failure.

4.3 Experimental Results

Fig.4 shows the center error distance changes over time. Fig.5 shows the SR at different thresholds ranging from 0 to 1. Fig.6 shows the PE plots at different thresholds ranging from 0 to 50. The numbers behind each tracker's name are also AUC values. Fig.7 shows the average SR and PE plots among all test sequences. Fig.8 shows some true tracking results of the compared trackers and our tracker, which are analyzed in details as follows.

Bike. In this sequence, the target bounding box contains a large background area, which influences classifier training. In this case, CT failed to distinguish the object from the background and drift in frame #63. Our tracker uses contour features to compare the similarity between particles and our trained template, which eliminates background disturbance. Even if the background area in bounding box is changing, the contour feature of the target is stable. We can see from Fig.4 that CSK, RPT, OAFT and our tracker evaluate the right object location and scale all the time.

*Car*4. In this sequence, the target car runs on a road and passes under several foot bridges. The difficulties of this video are the shadow and changing scale of the target. Fig.1 demonstrates that the edge and the layout of the car do not change much under different illumination conditions. As shown in Fig.8, all trackers except ours drift to the background when the car goes through foot bridge. It is because the contour features are robust to illumination shadow change (frames #195 and # 249). From Figs.4∼6, only our algorithm can track the car all the time and achieve the highest score beyond other trackers.

CarDark. The object car is running on the night and ill-lit street. In Fig.8, many trackers drift heavily due to poor light condition and complex background. Our tracker is designed to handle illumination with contour features. Therefore, even if light condition changes, the contour features remain stable. As shown in Figs.4∼6, OAFT and our method track the car steadily and achieve the highest scores.

David. In this sequence, a man walks around in a room. From time step #159 to time step #226, the target undergoes large appearance change due to scale and pose variation. Several trackers drift or track a wrong scale, but our tracker performs well. This is because our update strategy could adapt such smooth appearance change. As illustrated in Fig.5, TLD achieves the highest score, and the following trackers are RPT and ours.

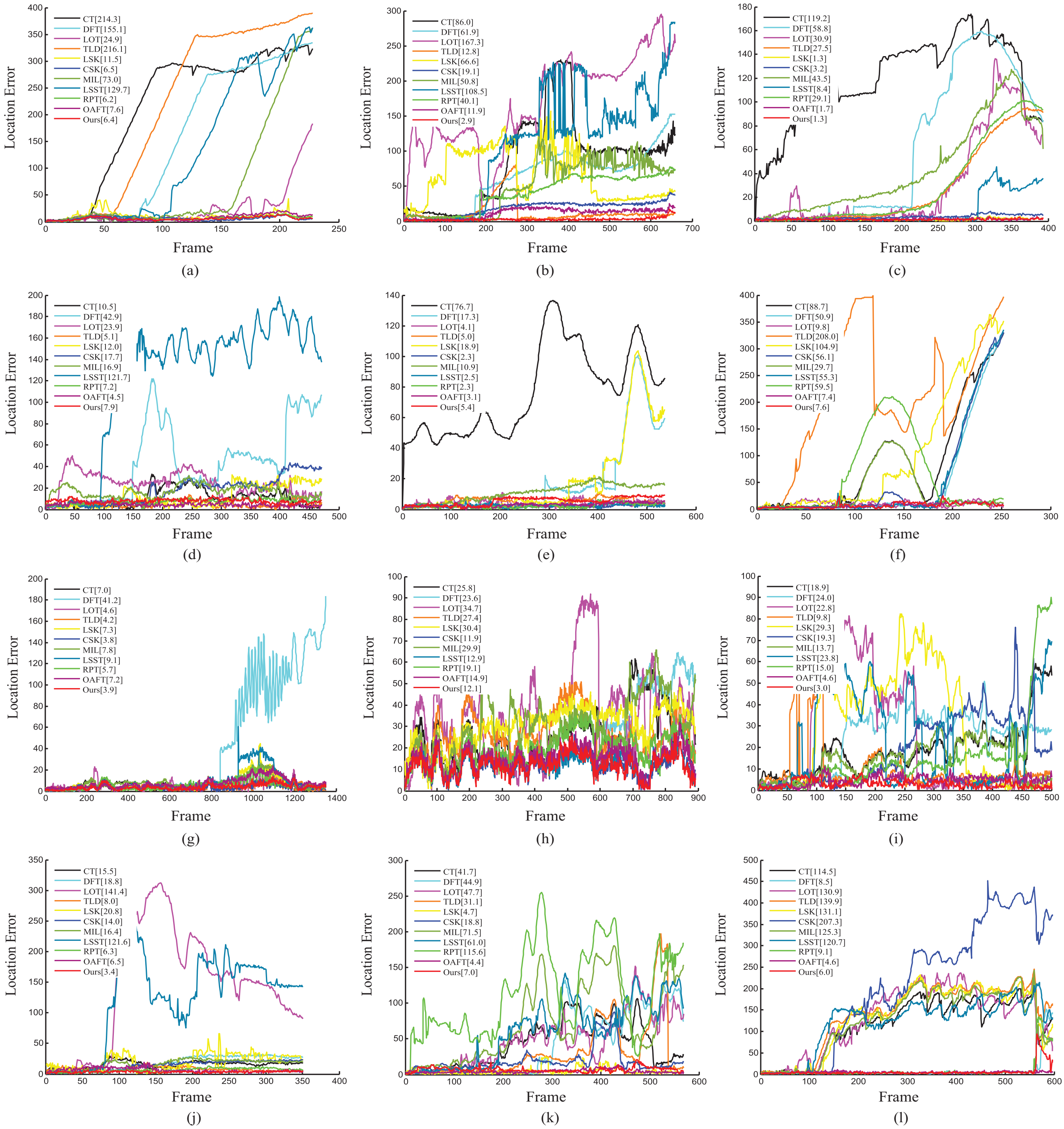
*David*2. In this sequence, we track a man in a room of which the background texture is complex. CT drifts to the background at the beginning. After the target moves into a region with complex background, LSK and DFT drift in frame #508. However, the proposed tracker does not drift in cluttered background. As illustrated in Fig.4, our tracker achieves a higher score (more than 0.9).

*David*3. In this sequence, the man is totally occluded by a tree twice during walking. As shown in Fig.8, when the man goes through the tree, RPT, CT, and MIL drift to the background. Other trackers, such as DFT, LSST and CSK, also drift after the man turned around. As we can see from Fig.5, our method achieves the highest score.

*Dog*1. In this sequence, the target is a toy dog of which the scale changes much. For example, from frame #910 to frame #1184, the bounding box becomes bigger when the target gets closer to the camera. Many trackers can track the correct location but wrong scale, such as CSK, MIL and CT. In our experiment, the samples are normalized into the same size before feature extraction step. Therefore although the scale changes, the contour feature of the target remains stable. As shown in Figs.4 and 5, LOT, LSK and our tracker can track the target in both correct location and scale.

*FaceOcc*1. In this sequence, a woman has taken a book in front of her face for a long time. As shown in Figs.5 and 6, all trackers drift to some degree because of heavy occlusion. Our tracker evaluates the similarity only using the bright part, which makes it robust to the occlusion problem (the same principle in Fig.3). As illustrated in Figs.4 and 5, CSK, LSST and our tracker achieve the highest score.

Girl. In this sequence, a girl sits on a chair with



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Fig.4. Center location errors. (a) Bike. (b) Car4. (c) CarDark. (d) David. (e) David2. (f) David3. (g) Dog1. (h) Faceocc1. (i) Girl. (j) Singer1. (k) Trellis. (l) Woman.

a camera. During frame #130 and frame #265, she moves her chair and turns herself a round. These actions cause scale change and self occlusion problem. Although the girl's head turns around, its contour features change small, which makes our tracker still able to locate it. As illustrated in Figs.5 and 6, we can see that only our tracker can track the girl well all the time, and the average score is far ahead of others.

*Singer*1. The object in the Singer1 sequence changes in scale and the illumination also changes tempestuously. If a tracker could not cope with illumination change, it is difficult to keep track of the object cor-

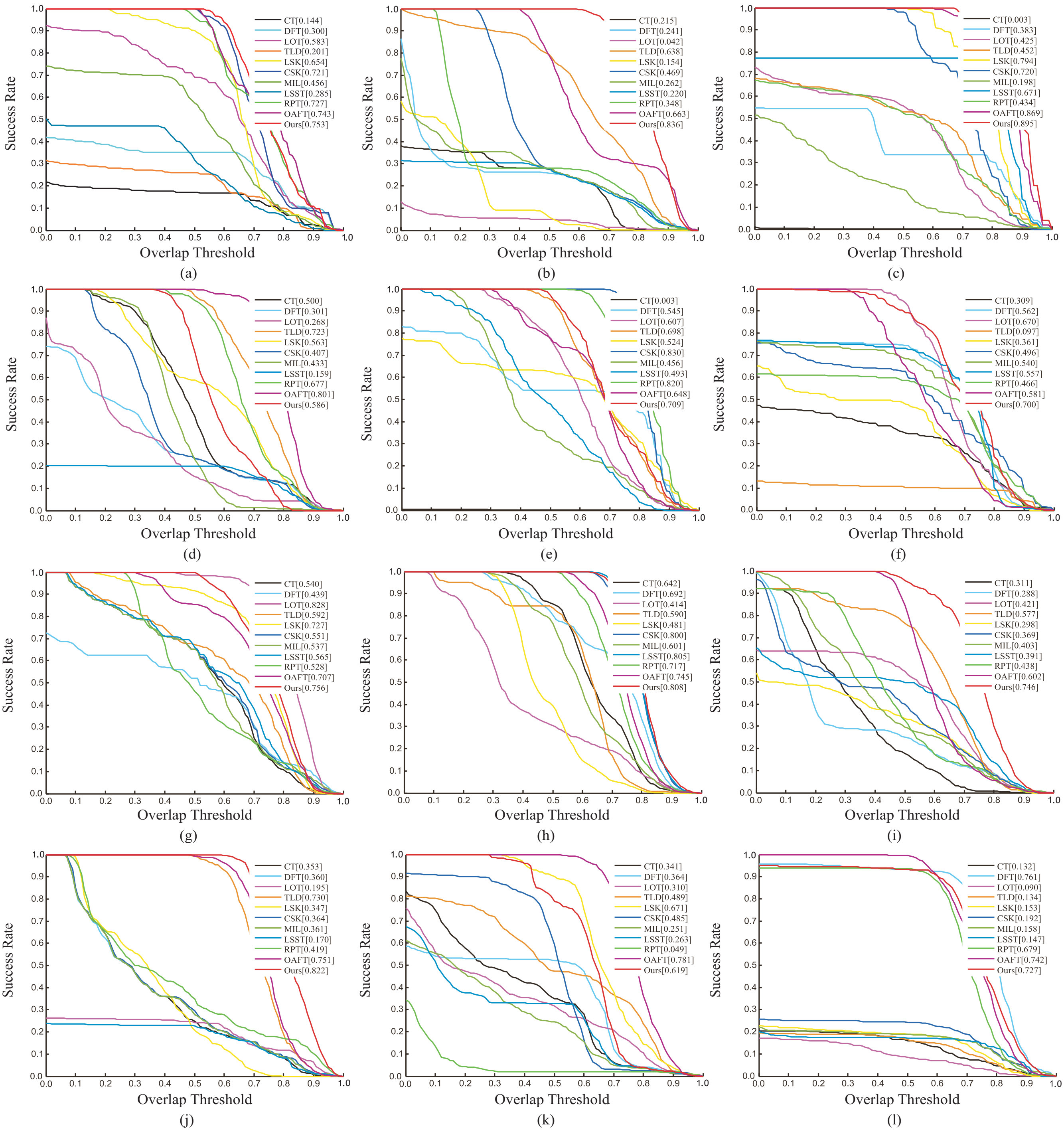


Fig.5. Overlap success rate plots. (a) Bike. (b) Car4. (c) CarDark. (d) David. (e) David2. (f) David3. (g) Dog1. (h) FaceOcc1. (i) Girl. (j) Singer1. (k) Trellis. (l) Woman.

rectly. Our algorithm is able to track the right object accurately in this sequence because it represents the target with illumination invariant features. As shown in Fig.5, TLD, OAFT and our tracker could track the target correctly in both location and scale.

Trellis. In this sequence, a man walks under a trellis and shadows on his face change heavily. Some trackers, such as RPT, MIL and LSST, drift because of shadows (frames #385, #476 and #540). Our appearance model involves illumination invariant contour features, which remain stable when the shadow changes. Therefore, our tracker can still track the target. It is shown

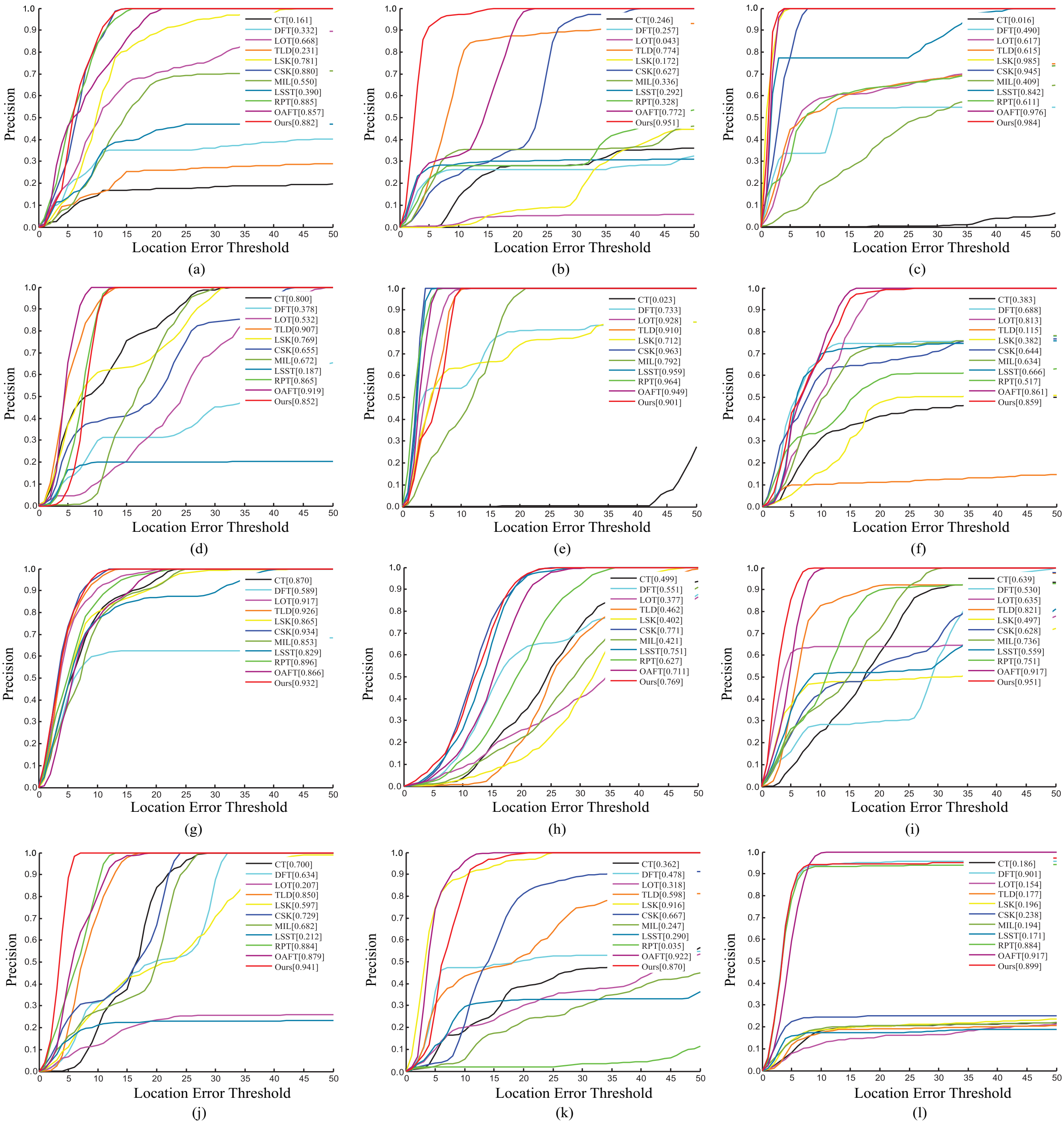
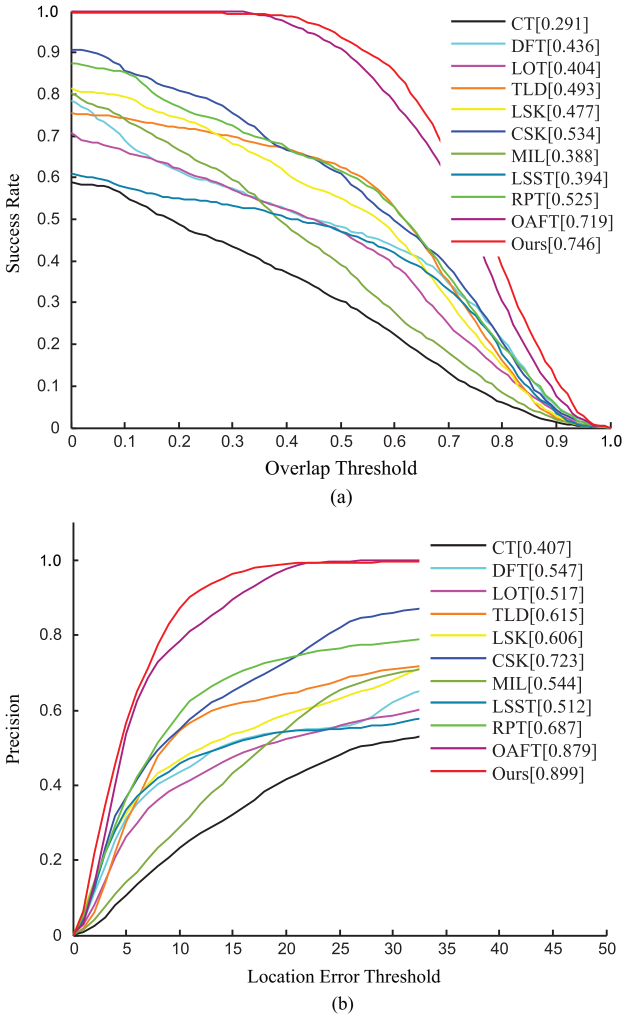


Fig.6. Precision plots by center location errors. (a) Bike. (b) Car4. (c) CarDark. (d) David. (e) David2. (f) David3. (g) Dog1. (h) FaceOcc1. (i) Girl. (j) Singer1. (k) Trellis. (l) Woman.

in Fig.4 that only LSK and our method can track the correct object location. As illustrated in Figs.5 and 6, our tracker achieves the second highest score.

Woman. In this sequence, a woman walks on a street and goes through some cars. From frame #111 to frame #165, when she walks on through the first white car, all the trackers drift except ours, RPT and DFT. Fig.3 explains why our tracker could deal with partial occlusion problem well. It is shown in Fig.4 that our tracker achieves the second highest score among all test trackers.

In a short, the experimental results demonstrate that the proposed tracker achieves a better performance than other trackers among these typical videos.



(a)

(b)

Fig.7. Average success and precision plots among all test sequences. (a) Success plots of TRE. (b) Precision plots of TRE.

5 Conclusions

This work constructed a set of complete contour features by convolving an image with two groups of well selected and integrated Prewitt and Sobel operators. The bright parts of contour features could provide more useful information than the dark parts. Therefore the bright pixels of the image are used. This strategy enables our algorithm to cope with partial occlusion problem. A number of experimental results showed that the proposed approach achieved competitive performance among state-of-the-art methods.

In the future, we will add deblurring ability. We will explore sophisticated filters and integrate them by using collaborative and parallel computing[38-40]. We also try to extend the idea into related areas of computer science[41-43].

References

[1] Wang N Y, Shi J P, Yeung D Y, Jia J Y. Understanding and diagnosing visual tracking systems. In *Proc. IEEE Int. Conf. Computer Vision*, December 2015, pp.3101-3109.

[2] Zhang X Q, Hu W M, Bao H J, Maybank S. Robust head tracking based on multiple cues fusion in the kernel-Bayesian framework. *IEEE Trans. Circuits and Systems for Video Technology*, 2013, 23(7): 1197-1208.

[3] Zhang X Q, Hu W M, Xie N H, Bao H J, Maybank S. A robust tracking system for low frame rate video. *International Journal of Computer Vision*, 2015, 115(3): 279-304.

[4] Zhang X Q, Hu W M, Qu W, Maybank S. Multiple object tracking via species-based particle swarm optimization. *IEEE Trans. Circuits and Systems for Video Technology*, 2010, 20(11): 1590-1602.

[5] Sun J, He F Z, Chen Y L, Chen X. A multiple template approach for robust tracking of fast motion target. *Applied Mathematics-A Journal of Chinese Universities*, 2016, 31(2): 177-197.

[6] Grabner H, Grabner M, Bischof H. Realtime tracking via on-line boosting. In *Proc. British Machine Vision Conf.*, September 2006, pp.47-56.

[7] Babenko B, Yang M H, Belongie S. Robust object tracking with online multiple instance learning. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2011, 33(8): 1619-1632.

[8] Hare S, Golodetz S, Saffari A, Vineet V, Cheng M M, Hicks S L, Torr P H S. Struck: Structured output tracking with kernels. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2016, 38(10): 2096-2109.

[9] Mei X, Ling H B. Robust visual tracking using ℓ_1 minimization. In *Proc. IEEE 12th Int. Conf. Computer Vision*, September 2009, pp.1436-1443.

[10] Li K, He F Z, Yu H P, Chen X. A correlative classifiers approach based on particle filter and sample set for tracking occluded target. *Applied Mathematics-A Journal of Chinese Universities*, 2017, 32(3): 294-312

[11] Zhong W, Lu H C, Yang M H. Robust object tracking via sparsity-based collaborative model. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2012, pp.1838-1845.

[12] Wu Y Q, He F Z, Zhang D J, Li X X. Service-oriented feature-based data exchange for cloud-based design and manufacturing. *IEEE Trans. Services Computing*, 2015, PP(99). doi: 10.1109/TSC.2015.2501981.

[13] Bao C L, Wu Y, Ling H B, Ji H. Real time robust L1 tracker using accelerated proximal gradient approach. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2012, pp.1830-1837.

[14] Ross D A, Lim J, Lin R S, Yang M H. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 2008, 77(1/2/3): 125-141.

[15] Kwon J, Lee K M. Visual tracking decomposition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2010, pp.1269-1276.

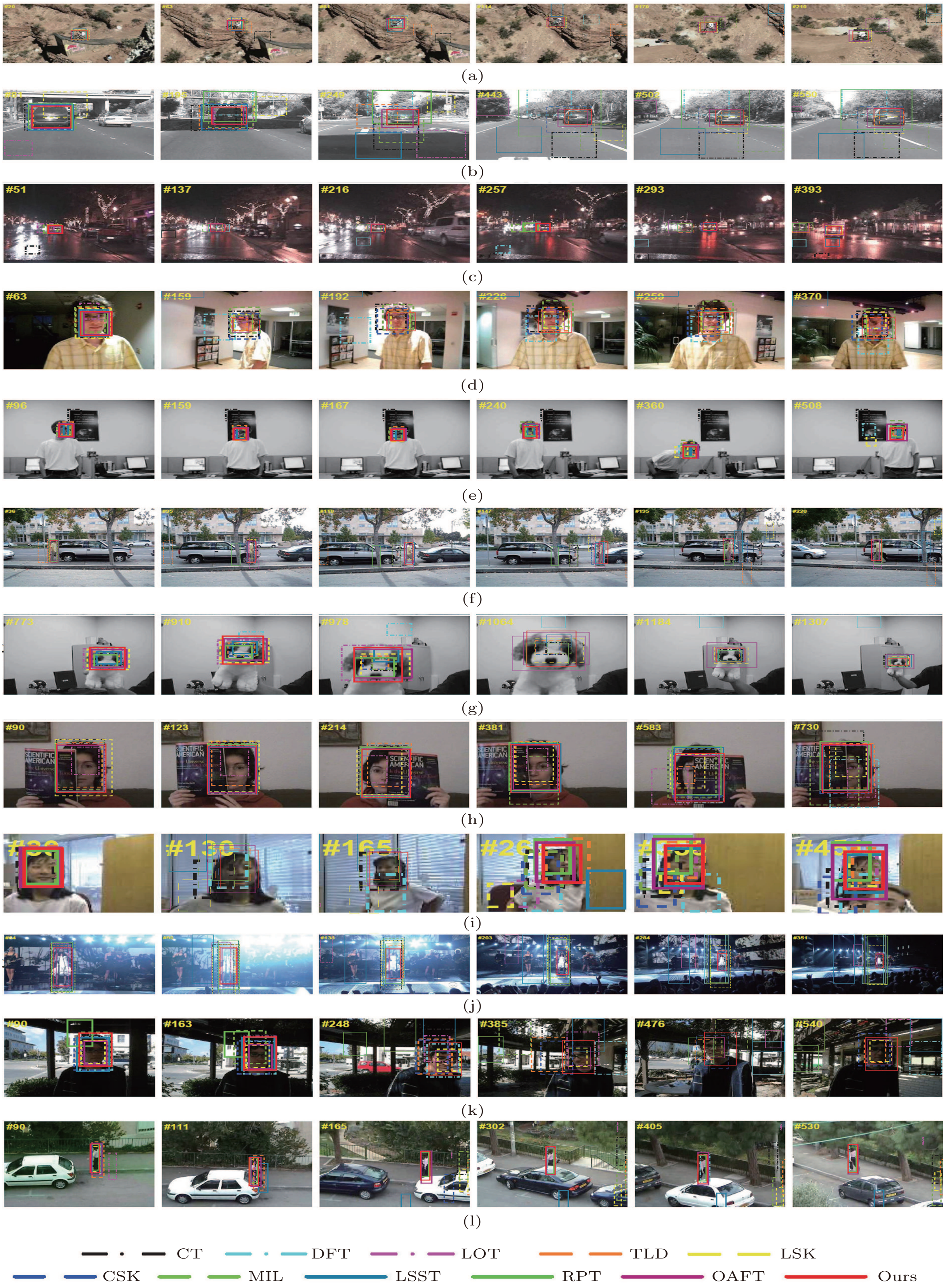

(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)
(k)
(l)

Fig.8. Some tracking results. (a) Bike. (b) Car4. (c) CarDark. (d) David. (e) David2. (f) David3. (g) Dog1. (h) FaceOcc1. (i) Girl. (j) Singer1. (k) Trellis. (l) Woman.

[16] Zhang K H, Zhang L, Yang M H. Real-time compressive tracking. In *Proc. the 12th European Conf. Computer Vision*, October 2012, pp.864-877.

[17] Ojala T, Pietikainen M, Mäenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2002, 24(7): 971-987.

[18] Li K, He F Z, Chen X. Real-time object tracking via compressive feature selection. *Frontiers of Computer Science*, 2016, 10(4): 689-701.

[19] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, December 2001, pp.I-511-I-518.

[20] Ni B, He F Z, Pan Y T, Yuan Z Y. Using shapes correlation for active contour segmentation of uterine fibroid ultrasound images in computer-aided therapy. *Applied Mathematics-A Journal of Chinese Universities*, 2016, 31(1): 37-52.

[21] Zhang D J, He F Z, Han S, Zou L, Wu Y Q, Chen Y L. An efficient approach to directly compute the exact Hausdorff distance for 3D point sets. *Integrated Computer Aided Engineering*, 2017, 24(3): 261-277.

[22] Chen Y L, He F Z, Wu Y Q, Hou N. A local start search algorithm to compute exact Hausdorff distance for arbitrary point sets. *Pattern Recognition*, 2017, 67: 139-148

[23] Li K, He F Z, Yu H, Chen X. A parallel and robust object tracking approach synthesizing adaptive Bayesian learning and improved incremental subspace learning. *Frontiers of Computer Science*. doi: 10.1007/s11704-018-6442-4

[24] Zhang D J, He F Z, Han S H, Li X X. Quantitative optimization of interoperability during feature-based data exchange. *Integrated Computer Aided Engineering*, 2016, 23(1): 31-50.

[25] Wang L, Liu T, Wang G, Chan K L, Yang Q X. Video tracking using learned hierarchical features. *IEEE Trans. Image Processing*, 2015, 24(4): 1424-1435.

[26] Ma C, Huang J B, Yang X K, Yang M H. Hierarchical convolutional features for visual tracking. In *Proc. IEEE Int. Conf. Computer Vision*, December 2015, pp.3074-3082.

[27] Wang L J, Ouyang W L, Wang X G, Lu H C. Visual tracking with fully convolutional networks. In *Proc. IEEE Int. Conf. Computer Vision*, December 2015, pp.3119-3127.

[28] Wolffsohn J S, Mukhopadhyay D, Rubinstein M. Image enhancement of real-time television to benefit the visually impaired. *American Journal of Ophthalmology*, 2007, 144(3): 436-440.

[29] Raheja J L, Kumar U. Human facial expression detection from detected in captured image using back propagation neural network. *International Journal of Computer Science & Information Technology*, 2010, 2(1): 116-123.

[30] Sevilla-Lara L, Learned-Miller E. Distribution fields for tracking. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2012, pp.1910-1917.

[31] Oron S, Bar-Hillel A, Levi D, Avidan S. Locally orderless tracking. *International Journal of Computer Vision*, 2015, 111(2): 213-228.

[32] Kalal Z, Mikolajczyk K, Matas J. Tracking-learning-detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2012, 34(7): 1409-1422.

[33] Liu B Y, Huang J Z, Yang L, Kulikowsk C. Robust tracking using local sparse appearance model and K-selection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2011, pp.1313-1320.

[34] Henriques J F, Caseiro R, Martins P, Batista J. Exploiting the circulant structure of tracking-by-detection with kernels. In *Proc. the 12th European Conf. Computer Vision*, October 2012, pp.702-715.

[35] Wang D, Lu H C, Yang M H. Least soft-threshold squares tracking. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2013, pp.2371-2378.

[36] Li Y, Zhu J K, Hoi S C H. Reliable patch trackers: Robust visual tracking by exploiting reliable patches. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2015, pp.353-361.

[37] Sun C, Wang D, Lu H C. Occlusion-aware fragment-based tracking with spatial-temporal consistency. *IEEE Trans. Image Processing*, 2016, 25(8): 3814-3825.

[38] Lv X, He F Z, Cai W W, Cheng Y. A string-wise CRDT algorithm for smart and large-scale collaborative editing systems. *Advanced Engineering Informatics*, 2017, 33: 397-409

[39] Zhou Y, He F Z, Qiu Y M. Optimization of parallel iterated local search algorithms on graphics processing unit. *The Journal of Supercomputing*, 2016, 72(6): 2394-2416.

[40] Zhou Y, He F Z, Qiu Y M. Dynamic strategy based parallel ant colony optimization on GPUs for TSPs. *Science China Information Sciences*, 2017, 60: 068102.

[41] Yan X H, He F Z, Hou N, Ai H J. An efficient particle swarm optimization for largescale hardware/software codesign system. *International Journal of Cooperative Information Systems*. doi: 10.1142/S0218843017410015.

[42] Cheng Y, He F Z, Wu Y Q, Zhang D J. Meta-operation conflict resolution for human-human interaction in collaborative feature-based CAD systems. *Cluster Computing*, 2016, 19(1): 237-253.

[43] Yan X H, He F Z, Chen Y L. A novel hardware/software partitioning method based on position disturbed particle swarm optimization with invasive weed optimization. *Journal of Computer Science and Technology*, 2017, 32(2): 340-355.

Kang Li is currently an assistant professor of School of Computer and Information Engineering of Hubei University, Wuhan. He received his B.S. degree in management from Anhui University, Hefei, in 2008. He received his M.S. degree in computer science from Huazhong Normal University, Wuhan, in 2012, and his Ph.D. degree in computer science from Wuhan University, Wuhan, in 2016. His research interests are computer vision, pattern recognition, image processing and computer graphics.

Fa-Zhi He is a professor of State Key Laboratory of Software Engineering, School of Computer Science, Wuhan University, Wuhan. He received his B.S., M.S. and Ph.D. degrees in computer science from Wuhan University of Technology, Wuhan, in 1990, 1996, and 2000 respectively. He was a postdoctoral researcher of Zhejiang University, a visiting researcher of Korea Institute of Science and Technology, and a visiting faculty member of the University of North Carolina at Chapel Hill. His research interests are computer-aided design, computer graphics, computer vision, image processing and collaborative computation.

Hai-Ping Yu is currently a Ph.D. candidate of State Key Laboratory of Software Engineering of Wuhan University, Wuhan. She received her B.S. degree in computer science and technology from Shandong Normal University, Jinan, in 2003. She received her M.S. degree in computer application technology from Wuhan University of Science and Technology, Wuhan, in 2005. Her research interests are computer vision, pattern recognition, image processing and computer graphics.