Multimed Tools Appl (2016) 75:5473–5492
DOI 10.1007/s11042-015-2516-6

Non-rigid visual object tracking using user-defined marker and Gaussian kernel

Guoheng Huang · Chi-Man Pun · Cong Lin · Yicong Zhou

Received: 14 July 2014 / Revised: 10 January 2015 / Accepted: 13 February 2015 /
Published online: 27 February 2015
© Springer Science+Business Media New York 2015

Abstract A novel non-rigid object tracking based on interactive user-define marker and superpixel Gaussian kernel is proposed in this paper. In the initialization stage, instead of using the traditional bounding box to locate the targeted object, we have employed an interactive segmentation with user-defined marker to segment the object accurately in the first frame of the input video to avoid the background influence in the traditional bounding box. During the tracking stage, by using a Gaussian kernel as movement constraint, each superpixel is tracked independently to locate the object in the next frame. Experimental results show that the proposed method compared to state of the art methods can achieve better robustness and accuracy for various challenging video clips.

Keywords Object tracking · Non-rigid · Superpixel · Interactive segmentation · Gaussian kernel

1 Introduction

Visual object tracking through video sequence is an important research area for a wide range of practical applications in computer vision recently including surveillance, driving assistant systems, augmented reality equipment, video segmentation and so on, which essentially deals with non-stationary data, both the target object and the background that change over time [33]. Recently, numerous algorithms have been proposed to address non-stationary data, and these algorithms have shown promising results [2–5, 8–10, 12, 14, 15, 19–21, 23, 26, 28, 30, 32, 34–36]. Ross et al. present an Incremental Learning Visual Tracker (IVT) that incrementally learns a low-dimensional subspace representation, efficiently adapting online to the appearance

G. Huang · C.-M. Pun (✉) · C. Lin · Y. Zhou
Department of Computer and Information Science, University of Macau, Macau, SAR, China
e-mail: cmpun@umac.mo

G. Huang
e-mail: yb27405@umac.mo

C. Lin
e-mail: yb17403@umac.mo

Y. Zhou
e-mail: yicongzhou@umac.mo

changes of the target [28]. For updating an adaptive appearance model of a tracking system, Babenko et al. [4] use Multiple Instance Learning (MIL) to train the appearance classifier results in more robust tracking, and presented an online boosting algorithm for MIL. Some algorithm for semi-supervised learning using a boosting framework (Semi Boost) [21] combines the advantages of graph based and ensemble methods, resulting in a better semi-supervised learning. Kahn et al. [36] replace the traditional importance sampling step in the particle filter with a novel Markov chain Monte Carlo (MCMC) sampling step to obtain a more efficient MCMC-based multi-target filter. They demonstrate that the resulting particle filters deal efficiently and effectively with complicated target interactions. Moreover, the template object is represented by multiple image fragments or patches in Frag-track [2]. In visual tracking decomposition (VTD) scheme [19], the observation model is decomposed into multiple basic observation models that are constructed by sparse principal component analysis of a set of feature templates. An on-line random forest (ORF) is proposed to overcome the limitation of traditional RF algorithms in practical usability [30].

In this paper, our main contributions are to propose a novel non-rigid object tracking scheme using interactive segmentation and superpixel Gaussian kernel. In the initialization stage, an interactive segmentation with user-defined marker is proposed to segment the object accurately in the first frame of the input video to avoid the background influence in the traditional bounding box. An over segmentation technique is applied to obtain certain number of superpixels, where the color and texture features are extracted by a color histogram and Contourlet transform. Then the object is located by the superpixel merging and the location matrixes are created for tracking. During the tracking stage, a Gaussian kernel is proposed as movement constraint, each superpixel is tracked independently to locate the object in the next frame. In the next section, the related works are described. In section 3, the initialization process for non-rigid tracking is described in details. In section 4, the superpixel based Gaussian kernel tracking algorithm will be explained. Experimental results are discussed in section 5. Finally, conclusion is drawn in section 6.

2 Related works

One of the most challenging visual tracking topics is non-rigid object tracking which handles the appearance variation of a target object. Though there are many difficulties, many works have been demonstrated to be effective or potential to track non-rigid objects in short durations and in well controlled environments. In this section, we have discussed the related non-rigid object tracking algorithms and put our work in proper context. A tracking method from the perspective of midlevel vision with structural information captured in superpixels is proposed in [11, 31], which is incorporated in an appearance model to distinguish the foreground target and the background. The appearance model is constructed by clustering a number of superpixels into different clusters. Patch-based method is one kind of most popular non-rigid object tracking algorithm. Kwon et al. develop a local patch-based appearance model and provide an efficient online updating scheme that adaptively changes the topology between patches [17, 18]. In the online update process, the robustness of each patch is determined by analyzing the likelihood landscape of the patch. Based on this robustness measure, the proposed method selects the best feature for each patch and modifies the patch by moving, deleting, or newly adding it over time. However, it is not enough to handle severe occlusions and multiple objects. Inspired by the patch-based algorithms, a novel coupled-layer visual model is proposed [6, 7]. It combines the target's global and local appearances by interlacing two layers. Mazinan and Amir-Latifi improve the Mean-shift (MS [10]) tracking algorithm by proposing an improved convex kernel function in association with the Kalman filter approach (KFA) [22]. It is able to estimate the location of the rigid and non-rigid objects. The below algorithms are successful in tracking non-rigid object. However, most of the existing non-rigid tracking approaches are limited by a bounding-box. Consequently, they would begin the tracker with a rather inaccurate object description in the first frame to include some background information. To avoid this problem, Godec put forward a generalized Hough transform based approach [13]. Especially, they apply the approach of Felzenszwalb to overcome the limitation of rectangular bounding-box [12].

Most related to [13], the initialization of the proposed algorithm is employed with an interactive segmentation scheme to segment the object accurately in the first frame of the input video to avoid the background influence in the traditional bounding box. Recently, semi-automatic segmentation methods incorporating simple user interaction have been actively researched [16, 24, 25, 27, 29]. The aim of interactive segmentation is to extract foreground objects from complex background through user interaction. In interactive segmentation, the user's interactive information is effectively employed for getting some prior information which leads to good segmentation performance. The MSRM algorithm ([24]) is most suitable for our paper, because our tracking scheme is also on region / superpixel level.

3 Object initialization using interactive segmentation

In our proposed non-rigid tracking scheme, the tracking process can be organized in two stages: initialization stage and tracking stage. In the initialization stage, an interactive segmentation is employed with user-defined marker to segment the object accurately in the first frame of the input video to avoid the background influence in the traditional bounding box. The traditional bounding-box initialization may not able to appropriately bound the object and results in large portion of background in the box [13]. The visual comparison is shown in Fig. 1, where the desired object is marked with green marker, and mark the background region with blue marker. From Fig. 1 (b) and (e), our initialization can accurately extract the desired target object in the first frame of the input video. Our approach allows tracking of objects with complicated background. However, as shown in Fig. 1 (c) and (f) using the traditional rectangular bounding box will include the irrelevant background information to represent the target object, which may cause less robust tracking performance.

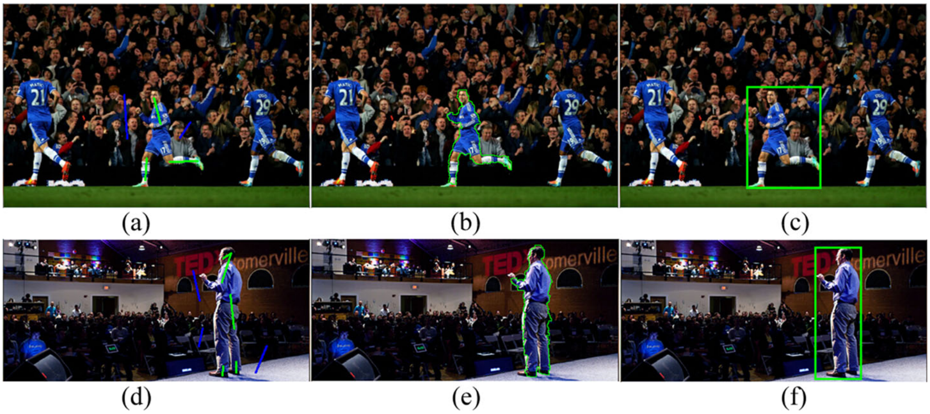


Fig. 1 Initialization in the first frame: (**a**) & (**d**) user marker; (**b**) & (**e**) interactive segmentation based initialization; (**c**) & (**f**) bounding-box initialization

In order to locate the target object accurately, interactive segmentation is employed with user-defined marker for object initialization. An over segmentation technique is applied to obtain certain number of superpixels, where the color and texture features are extracted by a color histogram and Contourlet transform. Then the object is located by the superpixel merging and the location matrixes are created for tracking. The overall the object initialization process is shown in Fig. 2 and the details of each step are described in following subsections.

3.1 Over segmentation

Although there are many over segmentation methods which can also generate superpixel for our proposed object initialization, we have chosen the fast and efficient Simple Linear Iterative Clustering (SLIC) [1] for superpixel generation. The SLIC method can divide an image into a number of equal-size superpixels and also better adhere to the object's edge. Therefore, the SLIC method is applied to the first frame of input video sequence to obtain m superpixels $\{s_{11}, s_{12}, \dots s_{1m}\}$.

3.2 Feature extraction

After the superpixel generation, the next step is to classify accurately the object contour and the background. One of the key issues in the following tracking process is how to determine the similarity distance between the unmarked superpixels with the marked object superpixels so that the candidate objects superpixels can be tracked with some logic control. Therefore, it is necessary to define a superpixel feature to measure the similarity distance between two superpixels s_i and s_j.

Our superpixel feature includes both color and texture information in order to handle the complex background. A superpixel is generally represented by a color histogram. However, we cannot rule out the case that superpixels belong to target object share close color with its neighboring background. Taking account of texture feature compensates the drawback of the sole color feature. Hence, on one hand, color histogram is applied as normalized color feature. In this paper, the RGB color space is used to compute the color histogram. We uniformly quantize each color channel into 16 levels and then the histogram of each region is calculated in the feature space of 16×16×16=4,096 bins [24]. Sum up, the normalized color feature is shown as below

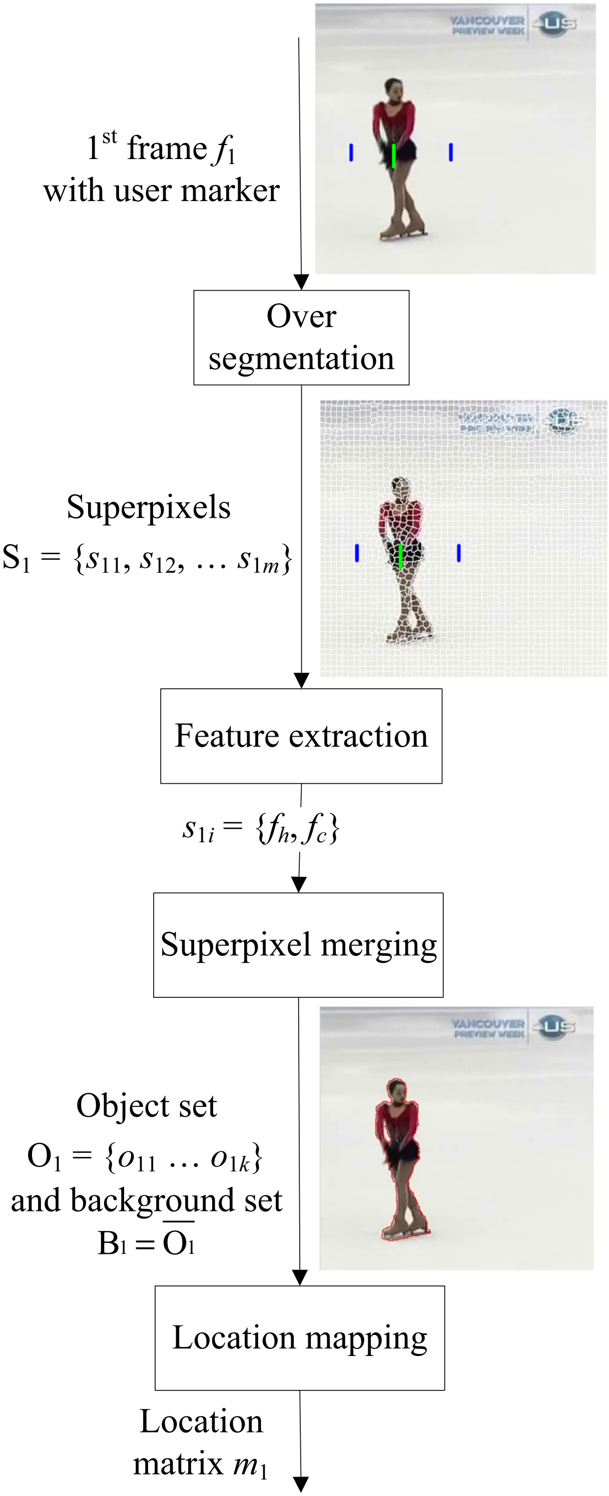
$$s_i.fh = \frac{(\mathrm{Hist}s_i{}^1, \dots, \mathrm{Hist}s_i{}^{4096})}{\mathrm{norm}((\mathrm{Hist}s_i{}^1, \dots, \mathrm{Hist}s_i{}^{4096}))} \tag{1}$$

where Hist_{si} is the normalized histograms of s_i, and the superscript u represents its uth element. On the other hand, the texture feature is a Contourlet transform based feature. First, every irregular superpixel is extended to its circumscribed square. In non-rigid object tracking, the target may rotate gradually and cause a tracking drift. To handle this challenge, the rotation invariant Contourlet feature is chosen as texture feature. In this paper, the Contourlet transform includes one level Wavelet and two levels of Pyramidal decomposition. Therefore, the normalized Contourlet transform based feature vector is as below

$$s_i.fc = \frac{(\mathrm{norm}(C_{i1}),\ \dots,\ \mathrm{norm}(C_{i10}))}{\mathrm{norm}((\mathrm{norm}(C_{i1}),\ \dots,\ \mathrm{norm}(C_{i10})))} \tag{2}$$

where i is the sequence number of each superpixel, and C_{ij}, j=1 … 10, is the component of Wavelet transform and two level Pyramidal decomposition. Then their normalized energy is computed to compose the 10-dimensional vector (norm (C_{i1}) … norm (C_{i10})).

Fig. 2 Framework of object initialization using interactive segmentation



3.3 Superpixel merging

Traditional tracking methods begin with a bounding-box which would cause less accurate foreground / background separation. Generally, the bounding-box is slightly larger than the object so that the object could be fully contained. However, due to the object is not always in rectangular shape, the bounding box unavoidably contained some background and the system mistakenly assumes that as parts of object. Therefore, we put forward to optimize the initial target extraction from the user input by interactive segmentation approach which is a region merging based method. It automatically merges the superixels segmented by initial segmentation, and then effectively extracts the object contour by labeling all the non-marker regions as either background or object.

First of all, the user input the superpixels of the first frame $\{s_{11}, s_{12}, \ldots s_{1m}\}$. Moreover, the user needs to mark the superpixels of interest and background. The color histogram and Contourlet transform based superpixel features of $\{s_{11}, s_{12}, \ldots s_{1m}\}$ are calculated to define the similarity of each superpixel. Then, object superpixels and background superpixels are respectively merge by maximal similarity based merging rule [24]. The whole MSRM process includes two stages which are repeatedly executed until there are two regions left – object and background. The scheme is to merge background regions as many as possible and keep object regions from being merged. Once all the background regions are merged, it means that the

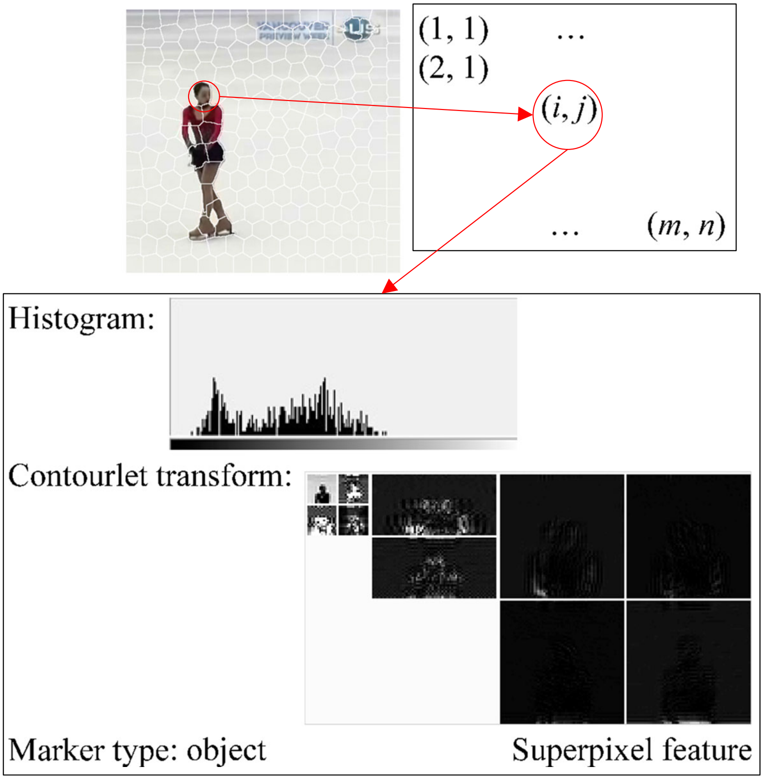


Fig. 3 Location mapping process: an image frame is mapped to superpixel matrix whose elements include histogram feature, Contourlet transform feature and marker type

target object is extracted. Finally, the object superpixel set $O_1=\{o_{11} \ldots o_{1m}\}$ and background superpixel set $B1 = \overline{O1}$ are extracted.

3.4 Location mapping

After SLIC initial segmentation and interactive segmentation based initial target extraction, the tracking will begin with the object and background superpixel sets. For more convenient and accurate to locate the object superpixels in next frame, map each frame of the input video into location matrixes.

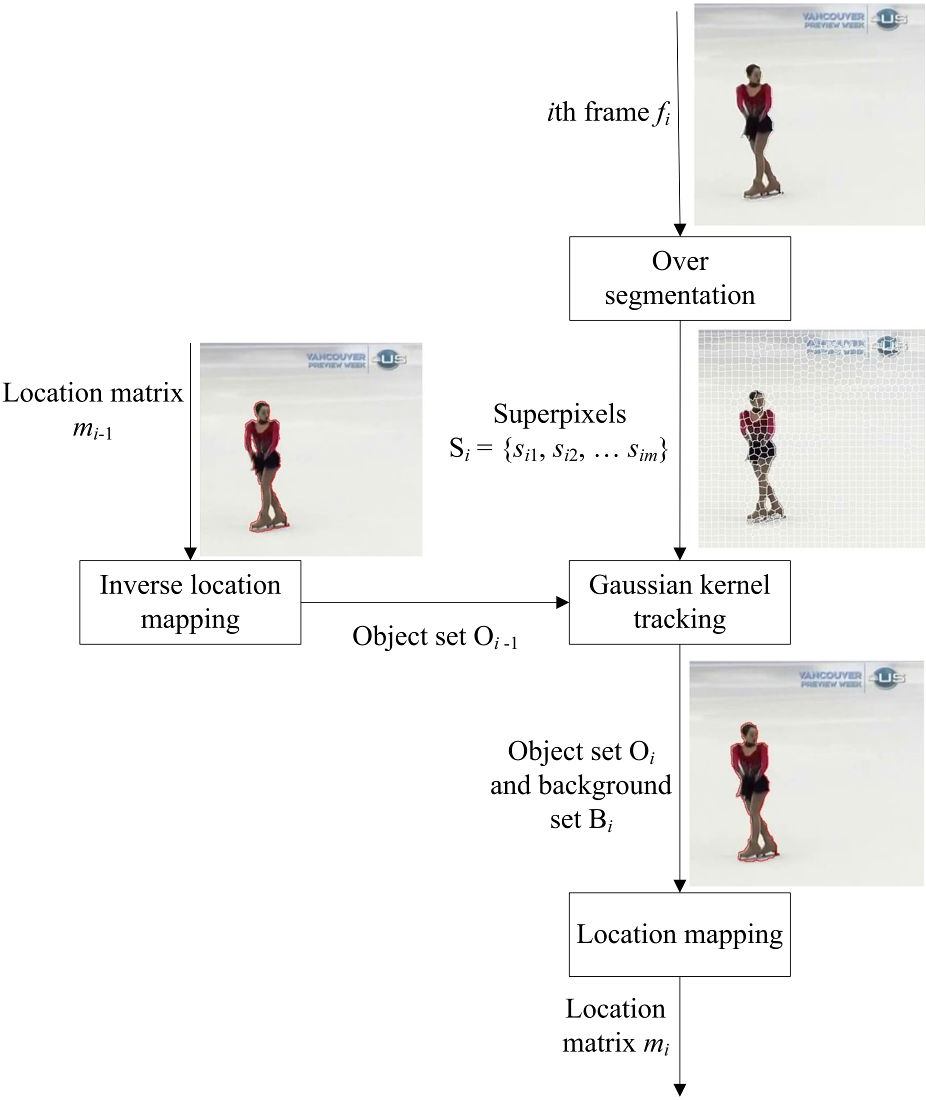

*i*th frame f_i
Over segmentation
Location matrix $m_{i\text{-}1}$
Superpixels $S_i = \{s_{i1}, s_{i2}, \ldots s_{im}\}$
Inverse location mapping
Gaussian kernel tracking
Object set $O_{i\text{-}1}$
Object set O_i and background set B_i
Location mapping
Location matrix m_i

Fig. 4 The proposed superpixel based Gaussian kernel tracking scheme

Every frame is transferred to a location matrix: each element in location matrix corresponds to each superpixel in the original frame; the corresponding location of each original superpixel will be represented by the coordinate of location matrix; the value of each element in location matrix is the superpixel feature of each superpixel in the original frames. As shown in Fig. 3, the current image frame is mapped to superpixel matrix whose elements include histogram feature, Contourlet transform feature and marker type. The marker type denotes the marker type of every superpixel. It includes three types: object, background and unknown.

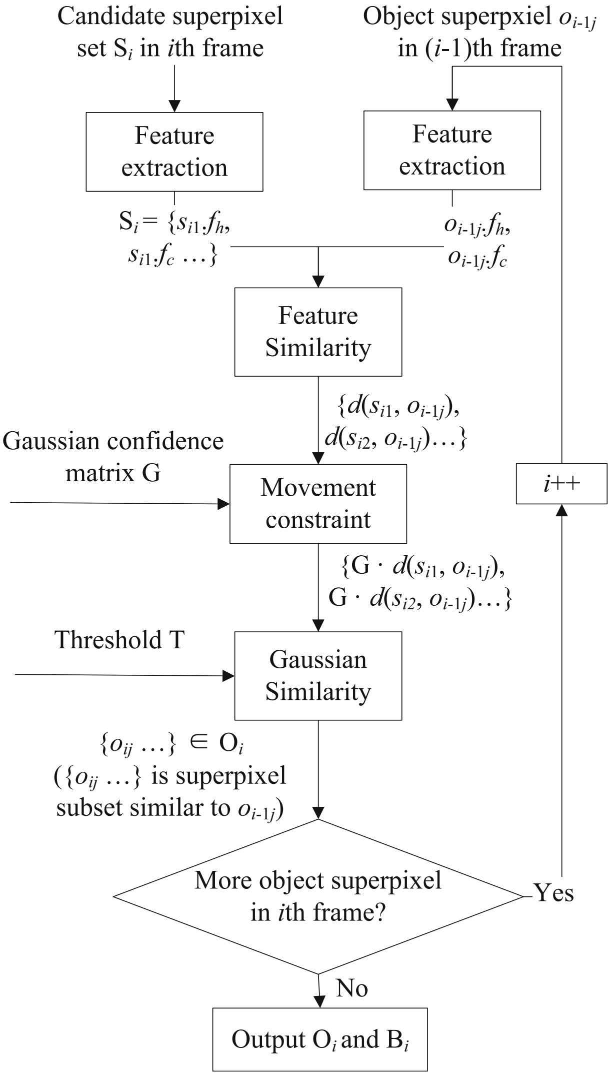
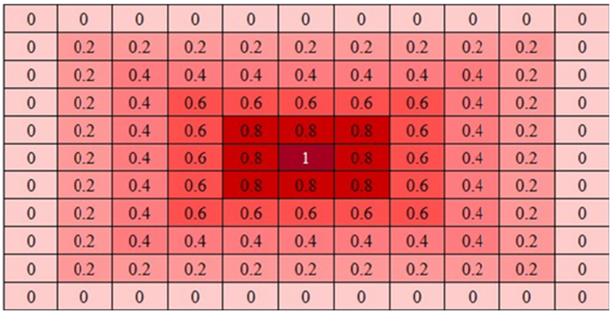

Candidate superpixel set S_i in ith frame
Object superpxiel $o_{i\text{-}1j}$ in (i-1)th frame
Feature extraction
Feature extraction
$S_i = \{s_{i1}.f_h, s_{i1}.f_c \ldots\}$
$o_{i\text{-}1j}.f_h, o_{i\text{-}1j}.f_c$
Feature Similarity
$\{d(s_{i1}, o_{i\text{-}1j}), d(s_{i2}, o_{i\text{-}1j})\ldots\}$
Gaussian confidence matrix G
Movement constraint
i++
$\{G \cdot d(s_{i1}, o_{i\text{-}1j}), G \cdot d(s_{i2}, o_{i\text{-}1j})\ldots\}$
Threshold T
Gaussian Similarity
$\{o_{ij} \ldots\} \in O_i$
($\{o_{ij} \ldots\}$ is superpixel subset similar to $o_{i\text{-}1j}$)
More object superpixel in ith frame?
Yes
No
Output O_i and B_i

Fig. 5 Gaussian kernel tracking for each superpixel in ith frame



0	0	0	0	0	0	0	0	0	0	0
0	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0
0	0.2	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.2	0
0	0.2	0.4	0.6	0.6	0.6	0.6	0.6	0.4	0.2	0
0	0.2	0.4	0.6	0.8	0.8	0.8	0.6	0.4	0.2	0
0	0.2	0.4	0.6	0.8	1	0.8	0.6	0.4	0.2	0
0	0.2	0.4	0.6	0.8	0.8	0.8	0.6	0.4	0.2	0
0	0.2	0.4	0.6	0.6	0.6	0.6	0.6	0.4	0.2	0
0	0.2	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.2	0
0	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0
0	0	0	0	0	0	0	0	0	0	0

Fig. 6 Explanation of Gaussian movement constraint through an example: the center is the location of object superpixel in previous frame; the number in each position is the Gaussian weight of the similarity between object superpixel in previous frame and candidate superpixel in current frame; the farther the superpixel is to the center, the less the Gaussian weight of similarity with the object superpixel is

4 Superpixel based Gaussian kernel tracking

The aim of tracking task is to find out the object superpixels in current frame using information in previous frames. In general, there are not dramatic changes in the similarity and probability density of two consecutive frames. Hence it could be reasonably assumed that a superpixel of the target in the previous frame does not move far away in the following frame. The object locations distanced away, which considered being less possible, should weigh much less than nearby locations. Therefore, a superpixel based Gaussian kernel tracking scheme is presented, where a Gaussian confidence map is employed as the movement constraint. The Gaussian based similarity measure between object superpixels in previous frame and superpixels in current frame depends on two factors: one is the feature of this superpixel, and the other is the confidence distance (Gaussian distance) between current superpixel in the current frame and the corresponding object superpixel in previous frame. Figure 4 shows the general framework for our proposed superpixel based Gaussian kernel tracking scheme. When an object superpixel in the (i-1)th frame and the candidate superpixels in the ith frame are input to the proposed superpixel based Gaussian kernel tracking scheme, every superpixel is represented by the superpixel feature based on color histogram and Contourlet transform. At the meantime, the similarity between the object superpixel and the candidate superpixels is measured. Then, the mentioned Gaussian kernel

Table 1 List of the parameters used in the experiments for our tracker

	Parameters
Over segmentation	size of superpixel: 144 compactness: 12
Feature similarity measure	w_1=0.8 w_2=0.2
Gaussian based movement constraint	σ=6.4
Gaussian similarity comparison	threshold T=0.0031

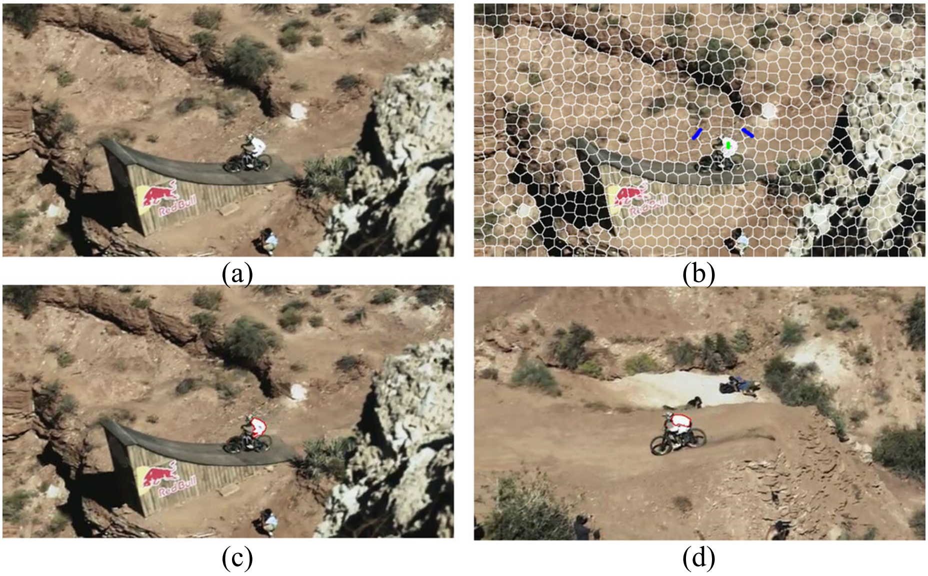


Fig. 7 The tracking process: (**a**) original frame (#1); (**b**) user marker and SLIC segmentation (#1); (**c**) interactive segmentation result (#1); (**d**) final tracking result (#227)

based movement constraint is applied to weigh the possible shift location of the successive object superpixel. Finally, after comparing the object superpixel with all the candidate superpixels, some candidate superpixels which are most similar to the object superpixel would be chosen as the object superpixels in the *i*th frame. This procedure will be repeated until all the remaining frames in video sequences has been processed.

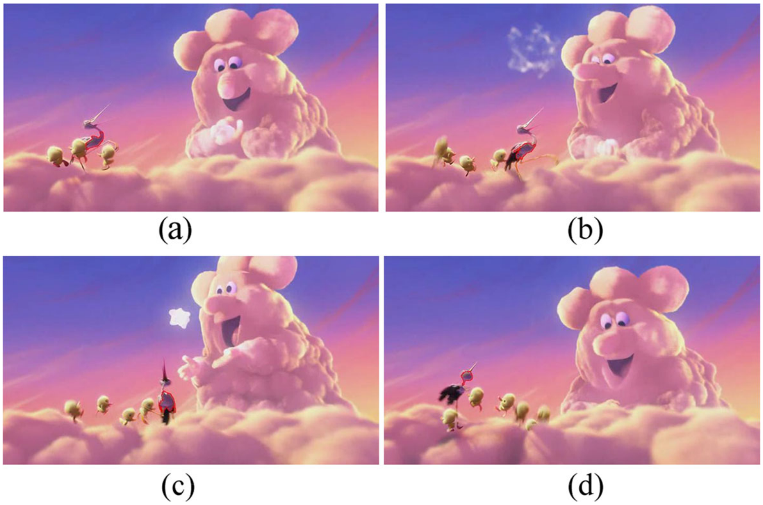
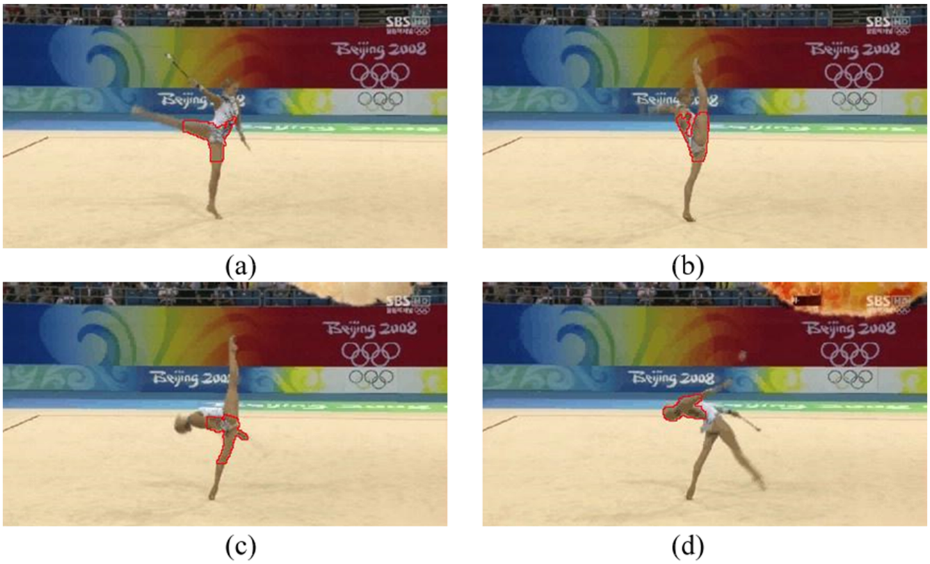


Fig. 8 Tracking result of our tracker on sequence *bird*: (**a**) frame #12; (**b**) frame #36; (**c**) frame #48; (**d**) frame #95

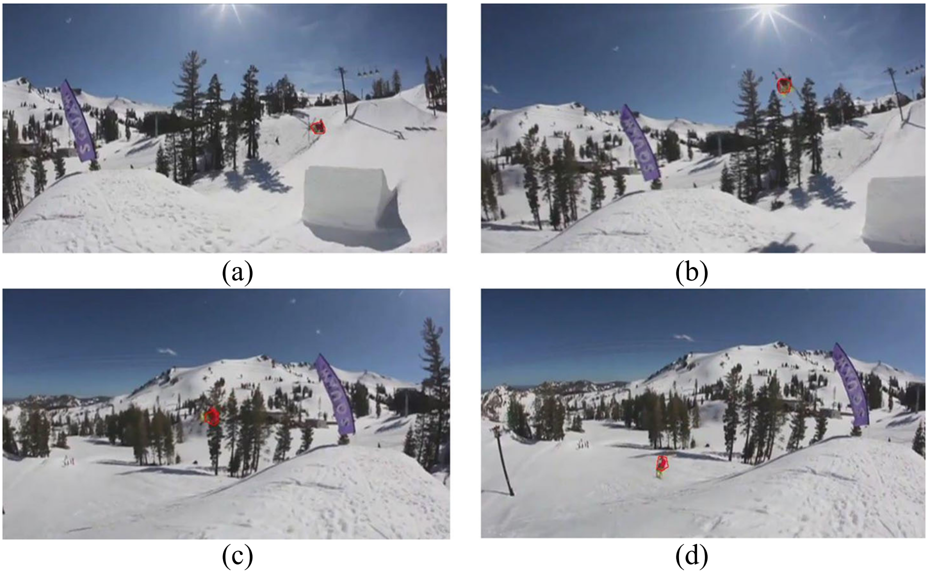


(a) (b) (c) (d)

Fig. 9 Tracking result of our tracker on sequence *gymnastics*: (**a**) frame #59; (**b**) frame #100; (**c**) frame #152; (**d**) frame #172

4.1 Feature similarity measure

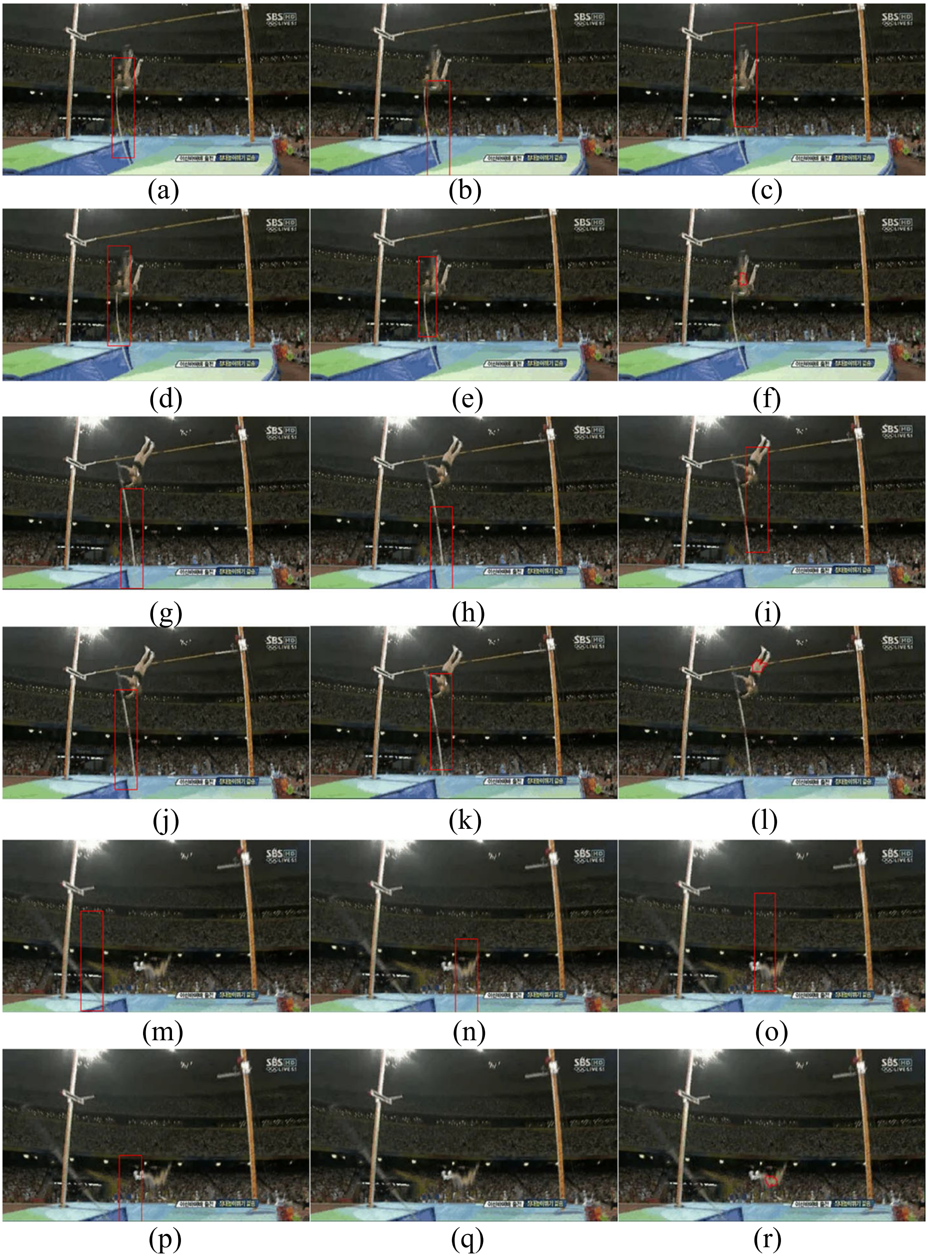
Suppose that s_i is the object superpixel in the previous frame, and s_j is one of candidate superpixels in the current frame. The superpixel similarity measure of color histogram d_h between s_i and s_j will be defined as $d_h(s_i, s_j)=f_h(s_i)\cdot f_h(s_j)$, which is within [0, 1]. The



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

Fig. 10 Tracking result of our tracker on sequence *skiing*: (**a**) Frame #3; (**b**) Frame #18; (**c**) Frame #41; (**d**) Frame #54

superpixel similarity measure of Contourlet transform d_c between s_i and s_j is defined as $d_c(s_i, s_j)=f_c(s_i)\cdot f_c(s_j)$, which is also within [0, 1]. Therefore, d is denoted by $d(s_i, s_j)=w_1\cdot d_h(s_i, s_j)+$


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

(p) (q) (r)

Fig. 11 Tracking results of five competing tracking algorithms and our tracker on sequence *high-jump*. (**a**) - (**f**) are the tracking results of CT, IVT, L1, MCMC, SPT and our tracker on #20. (**g**) - (**l**) are the tracking results on #31. (**m**) - (**r**) are the tracking results on #67 ((**q**) is the tracking result of SPT on #67, but the bounding box of tracking result is out of the frame)

$w_1 \cdot d_c(s_i, s_j)$, where w_1 and w_2 are weights. The higher d is, the higher the similarity between the features of two superpixels is. Figure 5 shows the Gaussian kernel tracking for each superpixel in ith frame.

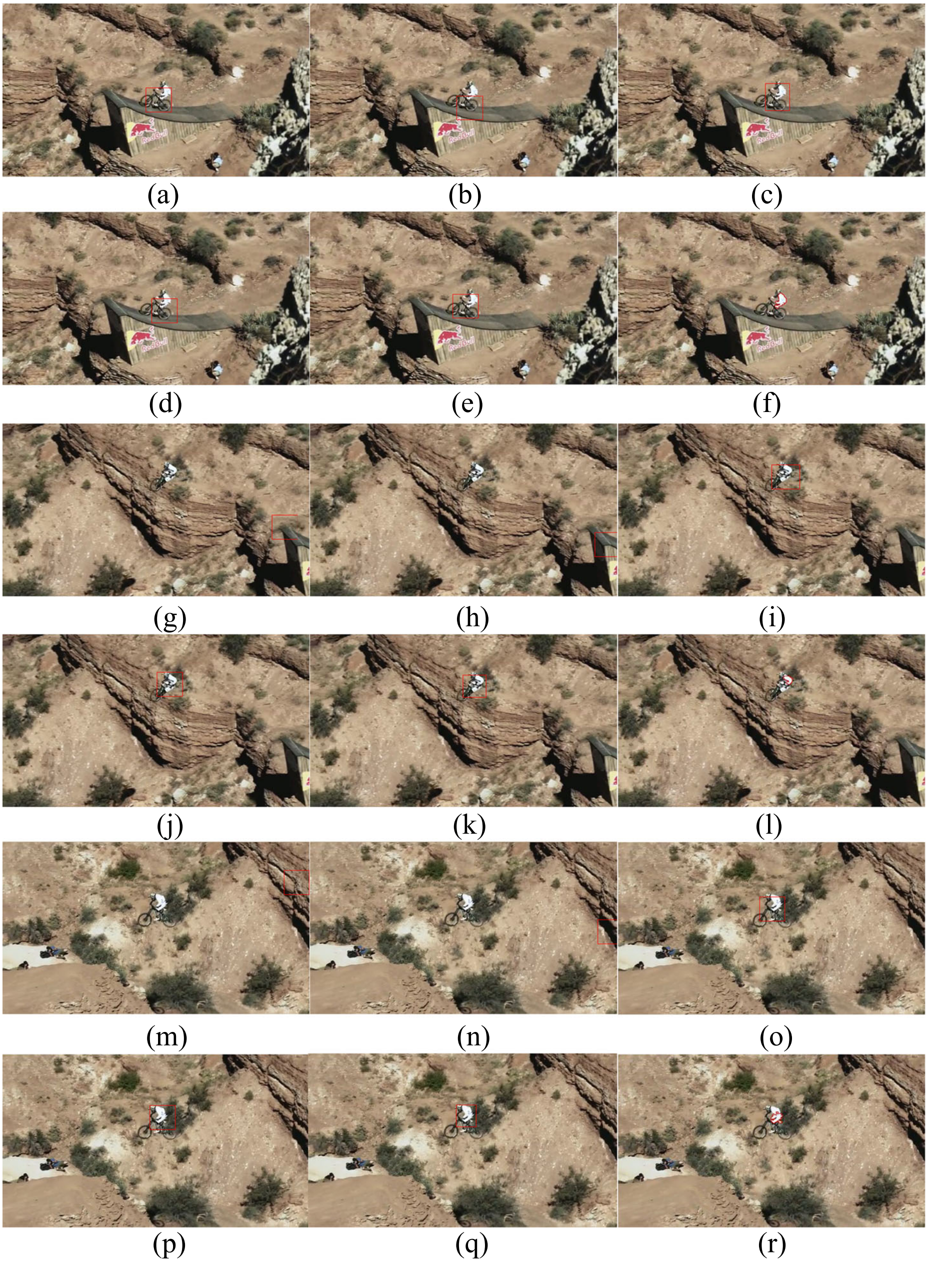
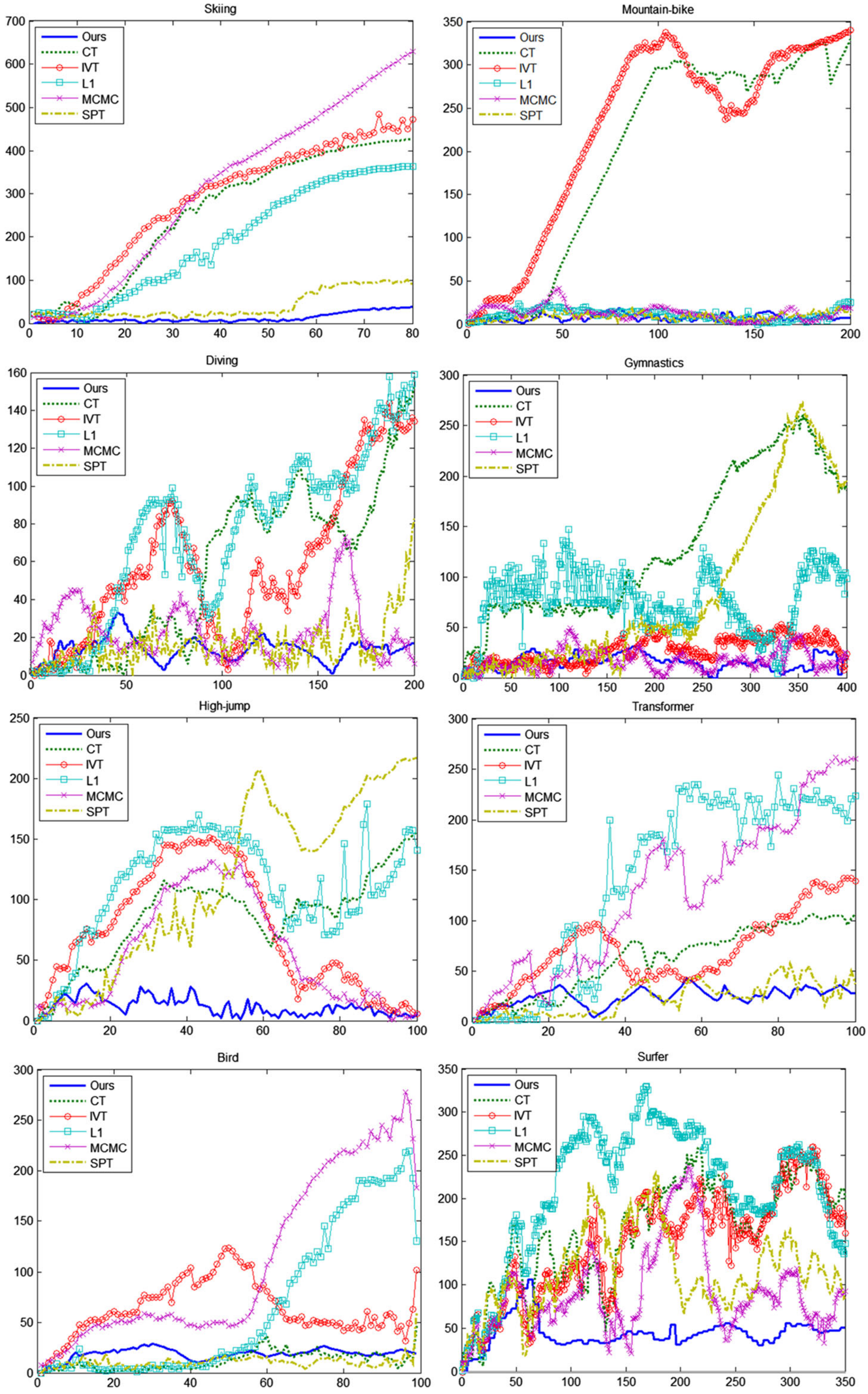


Fig. 12 Tracking results of five competing tracking algorithms and our tracker on sequence *mountain-bike*. (**a**) - (**f**) are the tracking results of CT, IVT, L1, MCMC, SPT and our tracker on #18. (**g**) - (**l**) are the tracking results on #94. (**m**) - (**r**) are the tracking results on #166


Skiing
Ours
CT
IVT
L1
MCMC
SPT
0
100
200
300
400
500
600
700
0
10
20
30
40
50
60
70
80
Mountain-bike
Ours
CT
IVT
L1
MCMC
SPT
0
50
100
150
200
250
300
350
0
50
100
150
200
Diving
Ours
CT
IVT
L1
MCMC
SPT
0
20
40
60
80
100
120
140
160
0
50
100
150
200
Gymnastics
Ours
CT
IVT
L1
MCMC
SPT
0
50
100
150
200
250
300
0
50
100
150
200
250
300
350
400
High-jump
Ours
CT
IVT
L1
MCMC
SPT
0
50
100
150
200
250
0
20
40
60
80
100
Transformer
Ours
CT
IVT
L1
MCMC
SPT
0
50
100
150
200
250
300
0
20
40
60
80
100
Bird
Ours
CT
IVT
L1
MCMC
SPT
0
50
100
150
200
250
300
0
20
40
60
80
100
Surfer
Ours
CT
IVT
L1
MCMC
SPT
0
50
100
150
200
250
300
350
0
50
100
150
200
250
300
350

◀ **Fig. 13** Tracking results comparison of CT, IVT, L1 tracker, MCMC, Superpixel tracking and our tracker. The horizontal axes in the sub-figures denote the number of frames in the video clips. The vertical axes in the sub-figures denote the Euclidean distance between the center of ground truth and segmented results at every frame (center error)

4.2 Gaussian based movement constraint

After the computation of the feature similarity measure for two superpixels, Gaussian kernel is applied to define the location difference between two superpixels. Formally, the Gaussian similarity of these two superpixels is defined as:

$$\mathrm{G}(\mathrm{center}x+\mathrm{distance}x,\ \mathrm{center}y+\mathrm{distance}y)\cdot d(si,sj) \tag{3}$$

where G is the Gaussian confidence matrix, (center*x*, center*y*) is the center of Gaussian kernel, distance*x* and distance*y* are the distance between s_i and s_j

$$\begin{aligned}\mathrm{distance}x&=\left|s_i.x{-}s_j.x\right|\\ \mathrm{distance}y&=\left|s_i.y{-}s_j.y\right|\end{aligned} \tag{4}$$

where $\mathrm{s}_i.x$ and $\mathrm{s}_j.x$ are the *x*-coordinate of the location of s_i and s_j; $\mathrm{s}_i.y$ and $\mathrm{s}_j.y$ are the *y*-coordinate of the location of s_i and s_j. Especially, the center of every Gaussian kernel map in current frame is the same location of every object superpixel in previous frame.

Therefore, for every object superixel in previous frame, the corresponding pending superpixel is new object superpixel in current frame when the value of $\mathrm{G}\cdot d$ is the greatest. More details about the movement constraint are shown by Fig. 6.

4.3 Gaussian similarity comparison

When the *i*th object superpixel s_i in previous frame is input, and s_i will be compared with the whole candidate superpixel set by using Gaussian similarity $\mathrm{G}\cdot d$. If a Gaussian similarity is greater than the predetermined threshold T, this candidate superpixel will be chosen as the corresponding object superpixel set to s_i. Therefore, the similar superpixels to s_i form an object superpixel subset. Then, the next object superpixel s_{i+1} will also compare with the remaining candidate superpixels. Likewise, the similar candidate superpixels to s_{i+1} will form another object superpixel subset in current frame. The iteration is over until find out all similar superpixels to object superpixel set in previous frame. By that time, the whole object superpixel set in current frame is collected.

5 Experimental results

We have analyzed the performance of our proposed non-rigid object tracking based on interactive segmentation and superpixel Gaussian kernel with several challenging video sequences. The sequences include either a non-rigid object which undergoes significant appearance changes. Our tracker is implemented in Matlab R2009a and runs on an Intel Core i7-3517U CPU with 4GB RAM. The tracker has been tested on some of the most challenging sequences (non-rigid case) are reported in this paper [11, 13, 31]. These sequences include complex background, fast movement, large variation in pose and scale, etc. The proposed algorithm have been compared with some prior works: standard Markov chain Monte Carlo based method (MCMC) [36], Incremental Learning Visual Tracker (IVT) [28], Compressive tracking (CT) [35], tracking using L1 minimization (L1) [32] and Superpixel tracking (SPT) [31]. In the following experiments, all algorithms start with the same target in the first frame. The parameters of our tracker are set as listed in Table 1.

In Fig. 7, we can comprehend the tracking process. After SLIC over segmentation, user needs to mark the desired target by user scribbles.

Since the intended purpose of our tracker is the tracking of non-rigid objects that may deform during runtime, the performance on several challenging sequences will be demonstrated later. Therefore, at first three sequences *bird*, *gymnastics* and *skiing* showing non-rigid deformations is evaluated, respectively consisting of about 103 frames, 766 frames and 80 frames. The results of Figs. 8, 9 and 10 show that our tracker can not only locate the objects' positions, but can also not contain any background region.

Figures 11 and 12 show the tracking results of other five competing algorithm and our tracker using sequence *high-jump* and *mountain-bike*. They are both challenging sequence where the non-rigid objects move dramatically. All other trackers are fail to track the object correctly in one sequence or both two sequences. On one hand, from Figs. 11 and 12, the results from our tracker are significantly better than the results of CT and IVT. Especially, the results of SPT might even be out the frame (Fig. 11 (q)). On the other hand, the background information outside the desired object is exclusive by our tracker. The test sequence *high-jump* includes the scale change of a target. In this case, the object became smaller during the tracking process. While our method is adaptively glued to the outline of target because of SLIC initial segmentation, and successfully tracked it.

The tracking results comparison of five competing algorithm and our tracker are presented in Fig. 13. It denotes the Euclidean distance between the center of ground truth and segmented results at every frame (center error) of different competing algorithms. The results show that our proposed method has better and robust performance for different video sequences.

Table 2 shows tracking results of the selected approaches evaluated on different video sequences. It denote the percentage of frames for each sequence until the tracking approach fails by visual inspection. In addition, Table 3 denote the average Euclidean distance from the center location of ground truth (average center error on the whole frame). From Table 2 and Table 3, our proposed tracker is more robust in different non-rigid tracking video sequences, while most other state-of-the-art methods fail in one or more video sequence.

Table 2 Quantitative evaluations 1: percentage of frames correctly tracked until failure

	CT [35]	L1 [32]	SPT [31]	IVT [28]	MCMC [36]	Ours
skiing	19	18	70	10	15	73
mountain-bike	22	100	100	11	100	100
diving	43	48	85	62	100	100
gymnastics	21	4	33	100	100	58
high-jump	28	15	28	15	28	100
transformer	34	29	100	52	31	100
bird	100	62	100	40	55	100
surfer	3	59	12	6	11	93
Average	34	42	66	37	55	91

Table 3 Quantitative evaluations 2: the numbers denote average errors of center location in pixels

	CT [35]	L1 [32]	SPT [31]	IVT [28]	MCMC [36]	Ours
skiing	256	119	42	284	312	12
mountain-bike	208	17	8	242	15	7
diving	72	87	24	71	21	14
gymnastics	135	67	148	31	16	28
high-jump	102	119	133	62	46	12
transformer	72	157	26	86	88	25
bird	12	65	11	115	75	15
surfer	158	203	112	144	98	46

6 Conclusion

In this paper, a novel non-rigid object tracking scheme using interactive user-defined marker and superpixel Gaussian kernel is proposed. By the combination of interactive segmentation techniques and superpixel based kernel tracking framework, it is able to track objects in some challenging non-rigid sequences. The interactive segmentation framework is employed for providing a better initialization of user input than traditional rectangle bounding-box based approaches. The interactive segmentation allows for an accurate initial separation of object and background to improve the accuracy of tracking. Moreover, experimental results demonstrate that our proposed method compared to state-of-the-art methods can achieve better robustness and accuracy for tracking non-rigid objects in different video sequences. For the future work, we would develop a more adaptive and effective over-segmentation to replace the SLIC over-segmentation in order to further improve the segmentation results of highly non-rigid objects.

Acknowledgments The authors would like to thank the referees for their valuable comments and Dr. Regina Chan for her proof reading of the manuscript. This research was supported in part by Research Committee of the University of Macau (MYRG134-FST11-PCM, MYRG181-FST11-PCM) and the Science and Technology Development Fund of Macau SAR (Project No. 008/2013/A1).

Contributions In this paper, Guoheng Huang and Chi-Man Pun are responsible for the design and writing, and implementation of the proposed method. Cong Lin and Yicong Zhou are responsible for the implementation, experiment settings and proof reading of the manuscript.

References

1. Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Susstrunk S (2012) SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Trans Pattern Anal Mach Intell 34(11):2274–2282
2. Adam A, Rivlin E, Shimshoni I (2006) Robust fragments-based tracking using the integral histogram. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 545–548
3. Avidan S (2007) Ensemble tracking. IEEE Trans Pattern Anal Mach Intell 29(2):261–271
4. Babenko B, Ming-Hsuan Y, Belongie S (2009) Visual tracking with online multiple instance learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 983–990
5. Bohyung H, Davis L (2005) On-line density-based appearance modeling for object tracking. In: Proceedings of IEEE international conference on computer vision, pp 1492–1499

6. Cehovin L, Kristan M, Leonardis A (2011) An adaptive coupled-layer visual model for robust visual tracking. In: Proceedings of IEEE international conference on computer vision, pp. 1363–1370
7. Cehovin L, Kristan M, Leonardis A (2013) Robust visual tracking using an adaptive coupled-layer visual model. IEEE Trans Pattern Anal Mach Intell 35(4):941–953
8. Collins RT, Yanxi L (2003) On-line selection of discriminative tracking features. In: Proceedings of IEEE international conference on computer vision, pp 346–352
9. Collins RT, Yanxi L, Leordeanu M (2005) Online selection of discriminative tracking features. IEEE Trans Pattern Anal Mach Intell 27(10):1631–1643
10. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 24(5):603–619
11. Fan Y, Huchuan L, Ming-Hsuan Y (2014) Robust superpixel tracking. IEEE Trans Image Process 23(4): 1639–1651
12. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE Trans Pattern Anal Mach Intell 32(9):1627–1645
13. Godec M, Roth P M, Bischof H (2011) Hough-based tracking of non-rigid objects. In: Proceedings of IEEE international conference on computer vision, pp 81–88
14. Han B, Zhu Y, Comaniciu D, Davis LS (2009) Visual tracking by continuous density propagation in sequential Bayesian filtering framework. IEEE Trans Pattern Anal Mach Intell 31(5):919–930
15. Jepson AD, Fleet D J, El-Maraghi TR (2001) Robust online appearance models for visual tracking. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp I-415-I-422
16. Jung C, Jian M, Liu J, Jiao L, Shen Y (2014) Interactive image segmentation via kernel propagation. Pattern Recogn 47(8):2745–2755
17. Junseok K, Kyoung Mu L (2013) Highly nonrigid object tracking via patch-based dynamic appearance modeling. IEEE Trans Pattern Anal Mach Intell 35(10):2427–2441
18. Junseok K, Kyoung-Mu L (2009) Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping Monte Carlo sampling. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1208–1215
19. Junseok K, Kyoung-Mu L (2010) Visual tracking decomposition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1269–1276
20. Kalal Z, Matas J, Mikolajczyk K (2010) P-N learning: Bootstrapping binary classifiers by structural constraints. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 49–56
21. Kumar Mallapragada P, Rong J, Jain AK, Yi L (2009) SemiBoost: boosting for semi-supervised learning. IEEE Trans Pattern Anal Mach Intell 31(11):2000–2014
22. Mazinan AH, Amir-Latifi A (2013) A new algorithm to rigid and non-rigid object tracking in complex environments. Int J Adv Manuf Technol 64(9–12):1643–1651
23. Ming Y, Ying W (2005) Tracking non-stationary appearances and dynamic feature selection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1059–1066
24. Ning J, Zhang L, Zhang D, Wu C (2010) Interactive image segmentation by maximal similarity based region merging. Pattern Recogn 43(2):445–456
25. Noma A, Graciano ABV, Cesar RM Jr, Consularo LA, Bloch I (2012) Interactive image segmentation by matching attributed relational graphs. Pattern Recogn 45(3):1159–1179
26. Nummiaro K, Koller-Meier E, Van Gool L (2003) An adaptive color-based particle filter. Image Vis Comput 21(1):99–110
27. Protiere A, Sapiro G (2007) Interactive image segmentation via adaptive weighted distances. IEEE Trans Image Process 16(4):1046–1057
28. Ross DA, Lim J, Lin RS, Yang MH (2008) Incremental learning for robust visual tracking. Int J Comput Vis 77(1–3):125–141
29. Rother C, Kolmogorov V, Blake A (2004) “GrabCut”: interactive foreground extraction using iterated graph cuts. ACM Trans Graph 23(3):309–314
30. Saffari A, Leistner C, Santner J, Godec M, Bischof H (2009) On-line random forests. In: Proceedings of IEEE international conference on computer vision, pp 1393–1400
31. Shu W, Huchuan L, Fan Y, Ming-Hsuan Y (2011) Superpixel tracking. In: Proceedings of IEEE international conference on computer vision, pp 1323–1330
32. Xue M, Haibin L (2011) Robust visual tracking and vehicle classification via sparse representation. IEEE Trans Pattern Anal Mach Intell 33(11):2259–2272
33. Yilmaz A, Javed O, Shah M (2006) Object tracking: a survey. ACM Comput Surv 38(4):13
34. Zhang X, Yu J, Wang T, Hou B, Jiao L C (2013) Path-based similarity with instance-level constraints for SemiBoost. In: Proceedings of SPIE international symposium on multispectral image processing and pattern recognition, pp 891911-891911-8

35. Zhang K, Zhang L, Yang MH (2012) Real-time compressive tracking. In: Proceedings of European conference on computer vision, pp 864–877
36. Zia K, Balch T, Dellaert F (2005) MCMC-based particle filtering for tracking a variable number of interacting targets. IEEE Trans Pattern Anal Mach Intell 27(11):1805–1819

Guoheng Huang received his B.Sc. degree in Applied Mathematics and M.E degree in Software Engineering from South China Normal University. He is currently a Ph. D. student majoring at software engineering at the University of Macau. His research interests include Image/Video Processing and Pattern Recognition.

Chi-Man Pun received his B.Sc. and M.Sc. degrees in Software Engineering from the University of Macau in 1995 and 1998 respectively, and Ph.D. degree in Computer Science and Engineering from the Chinese University of Hong Kong in 2002. He is currently an Associate Professor at the Department of Computer and Information Science of the University of Macau. He has investigated several funded research projects and published more than 100 refereed scientific papers in international journals, books and conference proceedings. Dr. Pun has served as the General Chair for the 10th International Conference Computer Graphics, Imaging and Visualization (CGIV2013), and program / session chair for several other international conferences. He has also served as the editorial member / referee for many international journals such as IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, Pattern Recognition, etc. His research interests include Digital Image Processing; Digital Watermarking; Pattern Recognition and Computer Vision; Intelligent Systems and Applications. He is also a senior member of the IEEE and a professional member of the ACM.

Cong Lin received his B.Sc. degree in Information & Computational Science from Guangdong University of Technology M.Sc. degree in Software Engineering from University of Macau. He is currently a Ph. D. student majoring at software engineering at the University of Macau. His research interests include Image/Video Processing and Pattern Recognition.

Yicong Zhou received his B.Sc. in from Hunan University, and M.Sc. and Ph.D. degrees in Electrical Engineering from the Tufts University, USA. He is currently an Assistant Professor at the Department of Computer and Information Science of the University of Macau. His research interests include Multimedia Security, Image/Signal Processing, Pattern recognition, Medical Imaging