Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Spatially constrained level-set tracking and segmentation of non-rigid objects ☆

Xuan Cheng [a], Ming Zeng [b], Xinguo Liu [a,*]

[a] State Key Lab of CAD&CG, Zhejiang University, China
[b] Software School of Xiamen University, China

ARTICLE INFO

Article history:
Received 28 November 2014
Revised 3 April 2016
Accepted 8 April 2016
Available online 27 April 2016

Keywords:
Segmentation-based tracking
Level-set
Hough voting
Back-projection

ABSTRACT

Level-set is a widely used technique in segmentation-based tracking due to its flexibility in handling 2D topological changes and computational efficiency. Most existing level-set models aim at grouping pixels that have similar features into a region, without consideration of the spatial relationship of these pixels. In this paper, we present a novel level-set tracking method that incorporates spatial information to improve the robustness and accuracy of tracking non-rigid objects. Both tracking and segmentation are performed in a unified probabilistic framework, with additional spatial constraints from a part-based model—the Hough Forests. In the stage of tracking, the rigid motion of the target object is estimated by rigid registration in both the color space and the Hough voting space. Then in the stage of segmentation, some support points are obtained from back-projection, and guide the level-set evolution to capture the shape deformation. We conduct quantitative evaluation on two recently proposed public benchmarks: a non-rigid object tracking dataset and the CVPR2013 online tracking benchmark, involving 61 sequences in total. The experimental results demonstrate that our tracking method performs comparably to the state-of-the-arts in the CVPR2013 benchmark, while shows significantly improved performance in tracking non-rigid objects.

1. Introduction

Recently segmentation-based tracking have attracted great attention in the field of object tracking. It could provide a more accurate foreground/background separation, compared with classical tracking methods which often use a bounding box to represent the target object. This is particularly important for tracking non-rigid objects, such as hands and pedestrians, because segmentation would introduce less undesirable background information and help avoid the drifting problem in a certain degree.

Level-set is a widely used technique in segmentation-based tracking, due to its flexibility in handing complex topological changes. Several methods have been proposed to track non-rigid or deformable objects in a level set framework. A representative work is the pixel-wise posteriors tracking [1], which performs level-set evolution and warping iteratively to track an object's contour and achieve some promising results. Then this work is extended to other tracking tasks, including multi-object tracking [2], pedestrians tracking [3,4] and 3D objects tracking [5]. Most of these methods rely on a global appearance model, such as color histograms, because of its convenience to describe the arbitrary shape of general object and computational efficiency. When it comes to tracking highly non-rigid objects in front of complex and cluttered backgrounds, the ability of a single global appearance model is relatively limited.

To capture the local property of objects, part-based representation is often used. For instance, the Deformable Part-based Model [6] is a prominent method in the domain of object detection. Some implicit part-based models can be obtained using the generalized Hough-transform [7], where each part of the object is mapped into a voting space in the Hough Forests framework. Furthermore, Godec et al. [8] extend the idea of Hough Forests to the online domain, and propose a Hough-based object tracking method.

The motivation of this paper is to integrate these part-based concepts into the level-set tracking. Compared with simple color histograms, part-based model such as Hough Forests, can provide mid-level information including the texture in local parts and the spatial constraints between these parts. According to previous literature, this information can benefit both the location [7] and the segmentation [8] of target object. However, it's not straightforward to utilize the part-based model in a level-set formalism, since

☆ This paper has been recommended for acceptance by M.T. Sun.

* Corresponding author.

E-mail addresses: chengxuan90@gmail.com (X. Cheng), xgliu@cad.zju.edu.cn (X. Liu).

http://dx.doi.org/10.1016/j.jvcir.2016.04.009

level-set usually operates on pixels directly while part-based model tends to handle local patches.

To address this issue, we present a probabilistic level-set framework, which uses Hough voting and support points provided by Hough Forests as the spatial constraints for tracking and segmentation. Begin with a generative model that utilizes color histograms and Hough Forests as the appearance model, we derive the formulation of tracking and segmentation respectively. In the stage of tracking, rigid registration is performed to make the new frame match the old frame in both color space and Hough voting space. Then in the stage of segmentation, we use back-projection to find the support points which have high confidence in belonging to the target object. These support points act as soft constraints for the subsequent level-set evolution. Finally, color histograms and Hough Forests are updated according to the segmentation results. Fig. 1 shows an example, and Fig. 2 depicts the overall procedure of our method.

The rest of this paper is organized as follows: Section 2 reviews some related works; Section 3 derives a probabilistic framework from a generative model; Section 4 outlines the tracking process; Section 5 shows the level-set segmentation; Section 6 presents online updating process; Section 7 shows some experimental results; Section 8 concludes this paper.

2. Related work

Object tracking methods could be categorized into two main classes, namely, bounding-box-based tracking and segmentation-based tracking. For bounding-box-based tracking, we refer readers to a comprehensive survey [9] and a recent benchmark [10]. Here we briefly review some representative works of segmentation-based tracking. Nejhum et al. [11] proposed to track articulated objects with a set of adaptively rectangular blocks, followed by a refine step using graph-cut segmentation. Fan et al. [12] introduced image matting into a tracking process, where the coarse tracking results provide suitable scribbles for matting, and both tracking and matting model are updated in closed-loop manner. Belagiannis et al. [13] combined tracking and segmentation in another way, where segmentation is used for sampling in the particle filtering framework. Godec et al. [8] extended the Hough Forest [7] into the online domain, which use a voting scheme to find the center of the object and back-project the pixels that voted for the object center to initialize the GrabCut segmentation. Later, Duffner and Garcia [14] proposed to use pixel-based descriptors instead of patch-based descriptors in the same online Hough voting scheme, and thus achieve faster tracking speed. Level-set technique, which implicitly represents the contours as the zero level-set of a higher dimensional function, is widely used in segmentation-based tracking. For example, Cremers [15] proposed a Bayesian level-set framework to track the contour of object and learn the dynamical shape priors simultaneously. Sun et al. [16] utilize the online boosting method as a detector to find the position of the object, and then obtain the contour with level-set. Bibby and Reid [1] derived a probabilistic level-set framework based on the pixel-wise posteriors. Their method comprises a rigid registration between frames, a segmentation and online appearance learning. Horbert et al. [4] improved on this work by using additional level-sets to enforce the spatial constraints of different parts of the object, in the context of pedestrians tracking. Our work is also built on [1], but we incorporate spatial constraints from Hough Forests to improve tracking performance.

3. A probabilistic level-set framework

Similar to [1], we present a probabilistic level-set framework for combined tracking and segmentation of an object. In this section, we firstly collect the notations used throughout this paper, and then emphasize two features of the Hough Forests model. We also describe a generative model that set the foundation of our proposed framework, and make some inferences to derive the formulations of tracking and segmentation.

3.1. Notations

Let $\mathbf{I}$ denote the image frame, and $\mathbf{I}_o$ denote the object frame (the black bounding box as shown in Fig. 3(a)). Let $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ denote the set of pixel locations in the object frame coordinate, and $\mathbf{y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ denote the set of corresponding pixel values. The object being tracked is represented by its shape $\mathbf{C}$, its position in the image $\mathbf{W}(\mathbf{x}, \mathbf{p})$, the color histograms M and the Hough Forests F. The shape is represented by the zero level-set $\mathbf{C} = \{\mathbf{x} | \mathbf{\Phi}(\mathbf{x}) = 0\}$ of an embedding function $\mathbf{\Phi}(\mathbf{x})$. The position is described by a warp $\mathbf{W}(\mathbf{x}, \mathbf{p})$ which warps a pixel location $\mathbf{x}$ in the object frame coordinate into the image frame coordinate according to parameters $\mathbf{p}$. The color histograms $M = \{M_f, M_b\}$ are built on foreground pixels and the nearby background pixels, with 32 bins per channel. The Hough Forests F contain two components that serve as spatial constraints, with F_v representing Hough voting map and F_s representing a set of support points.

3.2. Hough forests

Hough Forests have been proposed by Gall et al. [7] in the context of object detection. Because of the speed and robustness to noisy training data, Hough Forests inspired a series of extensions and applications in computer vision, such as human pose estimation from depth [17] and facial feature points detection [18]. Hough Forests are in fact a variant of the Implicit Shape Model [19], and thus use a star shaped model to represent the object, where each part of the object is connected to a centroid point through Hough voting procedure.

A training sample for Hough Forests is an image patch that consists of three elements: the feature of the patch, the foreground/background label of the patch and the offset vector pointing to expected object center. In the training process, several randomized tree structures are built to optimize the class impurity or the offset

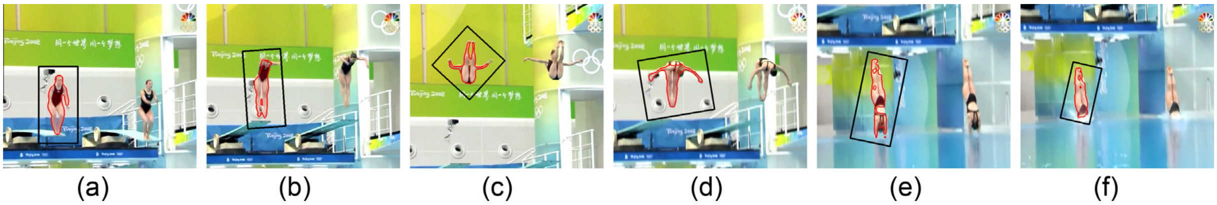

Fig. 1. Simultaneously tracking and segmentation of non-rigid object using our method. In tracking, a bounding box (black) is obtained by image warping with translation and rotation. In segmentation, a contour (red) is calculated by level-set evolution. The size of the bounding box is slightly updated to make it tighter to the target object. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

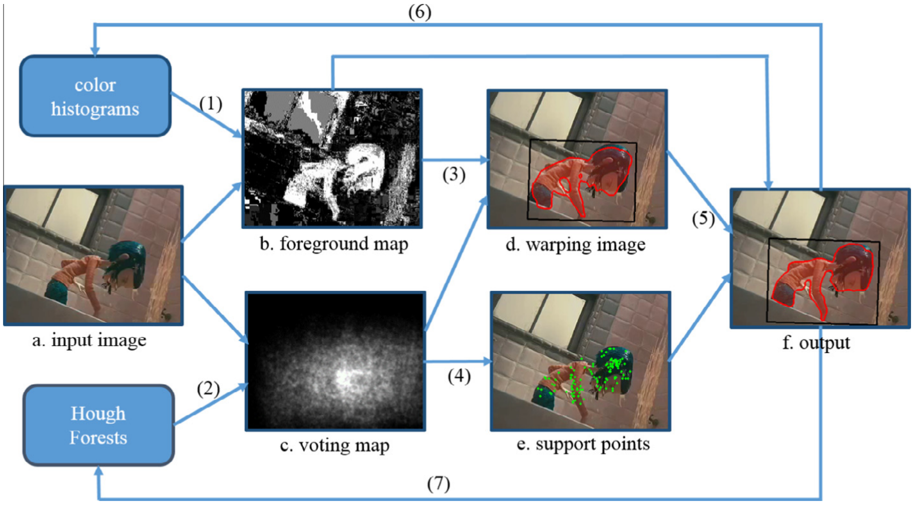

Fig. 2. The overall procedure for one frame. (1) When a new frame arrives, we firstly calculate its foreground/background probability using color histograms. (2) We also perform Hough voting for the new frame based on the Hough Forests model. (3) Then we estimate the pose of the target object through rigid registration. (4) The points that contribute to the center of the object are determined. (5) Next, a level-set evolution is performed to refine the shape of the object. (6 and 7) Finally, both color histograms and Hough Forests are updated using the segmentation results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

impurity. During evaluation, image patches extracted from all possible locations in the image go through the random trees in Hough Forests, and thus a voting map is generated by accumulating the voting vectors. After normalization, the intensity of the voting map on a specific position $\mathbf{x}_n$ corresponds to the probability of an object being centered there, denoted by $P(\mathbf{x}_n|F_v)$. This procedure is called *Hough voting*, which provides some useful spatial information for tracking.

Once we find the center of the object, either by calculating the maximum in the voting map [7,8] or through the rigid registration used in our method, we can obtain a set of *support points* F_s that contribute to voting the object center. This process is called back-projection [8]. Since the support points are more likely to belong to the object, they can be used as soft constraints $P(\mathbf{x}_n|F_s)$ in the following level-set evolution to help increase the robustness and accuracy.

3.3. The generative model

We now describe the probabilistic generative model which we use to derive the formulation of object tracking and segmentation, as shown in Fig. 3(b). The intuition behind the graphical model is that, given the shape $\mathbf{\Phi}$, the pose $\mathbf{p}$, the appearance models M and F, we can sample a pixel $\{\mathbf{x}_n, \mathbf{y}_n\}$ which tell us where the pixel is $\mathbf{x}_n$ and what color it has $\mathbf{y}_n$. According to this generative model, the joint distribution for a pixel is:

$$P(\mathbf{x}_n, \mathbf{y}_n, \mathbf{\Phi}, \mathbf{p}, M, F) = P(\mathbf{x}_n|\mathbf{\Phi}, \mathbf{p}, M, F)P(\mathbf{y}_n|M)P(M)P(F)P(\mathbf{\Phi})P(\mathbf{p}) \tag{1}$$

Firstly, we condition on $\mathbf{x}_n$ and $\mathbf{y}_n$, where both $P(\mathbf{x}_n)$ and $P(\mathbf{y}_n)$ are assumed to be constant. Then we condition on F, so the prior term $P(F)$ is removed. Dividing both sides by $P(\mathbf{x}_n)$, $P(\mathbf{y}_n)$ and $P(F)$, we get the following expression:

$$P(\mathbf{\Phi}, \mathbf{p}, M|\mathbf{x}_n, \mathbf{y}_n, F) = P(\mathbf{x}_n|\mathbf{\Phi}, \mathbf{p}, M, F)P(\mathbf{y}_n|M)P(M)P(\mathbf{\Phi})P(\mathbf{p}) \tag{2}$$

Secondly, we separate $P(\mathbf{x}_n|F)$ from conditional probability, since $\mathbf{\Phi}$, $\mathbf{p}$, M, F are independent with each other:

$$P(\mathbf{\Phi}, \mathbf{p}, M|\mathbf{x}_n, \mathbf{y}_n, F) \propto P(\mathbf{x}_n|\mathbf{\Phi}, \mathbf{p}, M)P(\mathbf{x}_n|F)P(\mathbf{y}_n|M)P(M)P(\mathbf{\Phi})P(\mathbf{p}) \tag{3}$$

Thirdly, we marginalize over M yielding the pixel-wise posterior probability of the shape $\mathbf{\Phi}$ and the position $\mathbf{p}$ given a pixel $\{\mathbf{x}_n, \mathbf{y}_n\}$ and the constraints form Hough Forests F:

$$P(\mathbf{\Phi}, \mathbf{p}|\mathbf{x}_n, \mathbf{y}_n, F) \propto \sum_{i=\{f,b\}} \{P(\mathbf{x}_n|\mathbf{\Phi}, \mathbf{p}, M_i)P(\mathbf{y}_n|M_i)\}P(\mathbf{x}_n|F)P(\mathbf{\Phi})P(\mathbf{p}) \tag{4}$$

Note that the prior terms $P(M_f)$ and $P(M_b)$ are dropped by assuming they are constant. Finally, we take product over the pixel-wise posterior

$$P(\mathbf{\Phi}, \mathbf{p}|\mathbf{x}, \mathbf{y}, F) \propto \prod_{n=1}^{N} \left\{ \sum_{i=\{f,b\}} \{P(\mathbf{x}_n|\mathbf{\Phi}, \mathbf{p}, M_i)P(\mathbf{y}_n|M_i)\} \right\} P(\mathbf{x}_n|F)P(\mathbf{\Phi})P(\mathbf{p}) \tag{5}$$

The goal of our method is to seek the parameters $\mathbf{\Phi}$, $\mathbf{p}$ that maximize the posterior probability $P(\mathbf{\Phi}, \mathbf{p}|\mathbf{x}, \mathbf{y}, F)$.

3.4. The probability distributions

Now we explain each of the distribution terms in Eq. (5) in detail:

- $P(\mathbf{x}_n|\mathbf{\Phi}, \mathbf{p}, M_i)$ is the probability of the pixel location $\mathbf{x}_n$ given the shape $\mathbf{\Phi}$, the pose $\mathbf{p}$, the color histogram M_i:

$$P(\mathbf{x}_n|\mathbf{\Phi}, \mathbf{p}, M_f) = H_\epsilon(\mathbf{\Phi}(\mathbf{x}_n)), \tag{6}$$

$$P(\mathbf{x}_n|\mathbf{\Phi}, \mathbf{p}, M_b) = 1 - H_\epsilon(\mathbf{\Phi}(\mathbf{x}_n)), \tag{7}$$

  where $H_\epsilon(z)$ is the smoothed Heaviside step function. The function of this term is to select foreground or background region.
- $P(\mathbf{y}_n|M_f), P(\mathbf{y}_n|M_b)$ represent the probabilities of the color $\mathbf{y}_n$ belonging to foreground and background, given the color histograms. $P(\mathbf{y}_n|M_f)$ and $P(\mathbf{y}_n|M_b)$ are normalized such that $P(\mathbf{y}_n|M_f) + P(\mathbf{y}_n|M_b) = 1$.
- $P(\mathbf{x}_n|F)$ represents the constraints from Hough Forests F on the pixel position $\mathbf{x}_n$. This term has different forms in tracking and segmentation, and it will be specified later (in Sections 4 and 5).
- $P(\mathbf{\Phi})$ is a prior term [4] on shape $\mathbf{\Phi}$. It has two functions: rewarding a signed distance function and smoothing the contour.

$$P(\mathbf{\Phi}) = \prod_{n=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \times \exp\left(-\frac{(|\nabla\mathbf{\Phi}(\mathbf{x}_n)| - 1)^2}{2\sigma^2}\right) \exp(-\lambda|\nabla H_\epsilon(\mathbf{\Phi}(\mathbf{x_n}))|) \tag{8}$$

where σ and λ are the weights.

- $P(\mathbf{p})$ is prior term on the pose $\mathbf{p}$. It's handled by drift correction as in [1], so we can drop it for brevity.
  Substituting all of the above into Eq. (5) and taking logs, we arrive at

$$\log(P(\mathbf{\Phi},\mathbf{p}|\mathbf{x},\mathbf{y},F)) \propto \sum_{n=1}^{N} \left\{\log(P(\mathbf{x}_n|\mathbf{\Phi},\mathbf{p},M)P(\mathbf{y}_n|M)) + \log(P(\mathbf{x}_n|F)) - \frac{(|\nabla\mathbf{\Phi}(\mathbf{x}_n)| - 1)^2}{2\sigma^2} - \lambda|\nabla H_\epsilon(\mathbf{\Phi}(\mathbf{x}_n))|\right\} \tag{9}$$

where

$$P(\mathbf{x}_n|\mathbf{\Phi},\mathbf{p},M)P(\mathbf{y}_n|M) = H_\epsilon(\mathbf{\Phi}(\mathbf{x}_n))P(\mathbf{y}_n|M_f) + (1 - H_\epsilon(\mathbf{\Phi}(\mathbf{x}_n)))P(\mathbf{y}_n|M_b) \tag{10}$$

To maximize Eq. (9) with respect to shape $\mathbf{\Phi}$ and pose $\mathbf{p}$, we use an alternating optimization method. First, we optimize $\mathbf{p}$ and keep $\mathbf{\Phi}$ fixed (tracking step). Then we optimize $\mathbf{\Phi}$ while keeping $\mathbf{p}$ constant (segmentation step).

4. Tracking

As the shape $\mathbf{\Phi}$ is fixed during tracking, we drop the prior term of shape $P(\mathbf{\Phi})$:

$$\log(P(\mathbf{\Phi},\mathbf{p}|\mathbf{x},\mathbf{y},F)) \propto \sum_{n=1}^{N}\{\log(P(\mathbf{x}_n|\mathbf{\Phi},\mathbf{p},M)P(\mathbf{y}_n|M)) + \log(P(\mathbf{x}_n|F))\}, \tag{11}$$

where the Hough Forest term is defined as follow:

$$P(\mathbf{x}_n|F) = H_\epsilon(\mathbf{\Phi}(\mathbf{x}_n))P(\mathbf{x}_n|F_v). \tag{12}$$

Here F_v represents the voting map, which is generated by Hough voting process. Hence, $P(\mathbf{x}_n|F_v)$ is the probability of the target object being centered on the position $\mathbf{x}_n$. Tracking is done by performing a rigid registration between current frame and new arriving frame, similar to the inverse compositional image alignment [20]. Through warping the new frame such that its content best fits current contour, we can find the position of the object in the new frame. Hence, we introduce a warp $\mathbf{W}(\mathbf{x},\Delta\mathbf{p})$ into Eq. (11), and further expand it:

$$\log(P(\mathbf{\Phi},\mathbf{p}|\mathbf{x},\mathbf{y},F)) \propto \sum_{n=1}^{N}\{\log\left(H_\epsilon(\mathbf{\Phi}(\mathbf{W}(\mathbf{x}_n,\Delta\mathbf{p})))P(\mathbf{y}_n|M_f) + (1 - H_\epsilon(\mathbf{\Phi}(\mathbf{W}(\mathbf{x}_n,\Delta\mathbf{p}))))P(\mathbf{y}_n|M_b)\right) + \log\left(H_\epsilon(\mathbf{\Phi}(\mathbf{W}(\mathbf{x}_n,\Delta\mathbf{p}))P(\mathbf{x}_n|F_v)\right)\}. \tag{13}$$

where $\Delta\mathbf{p}$ represents an incremental warp.

Similar to [1], we use a Gauss–Newton scheme to maximize Eq. (13) with respect to $\Delta\mathbf{p}$. All terms in Eq. (13) are strictly positive and therefore can be written as squared square-roots. Each square-root is approximated with a first-order Taylor series, with $h = H_\epsilon(\mathbf{\Phi}(\mathbf{x}_n))$:

$$H_\epsilon(\mathbf{\Phi}(\mathbf{x}_n)) = \left[\sqrt{H_\epsilon(\mathbf{\Phi}(\mathbf{x}_n))}\right]^2 \approx \left[\sqrt{h} + \frac{1}{2\sqrt{h}}\mathbf{J}\Delta\mathbf{p}\right]^2, \tag{14}$$

and similarly

$$1 - H_\epsilon(\mathbf{\Phi}(\mathbf{x}_n)) = \left[\sqrt{1 - H_\epsilon(\mathbf{\Phi}(\mathbf{x}_n))}\right]^2 \approx \left[\sqrt{1-h} - \frac{1}{2\sqrt{1-h}}\mathbf{J}\Delta\mathbf{p}\right]^2, \tag{15}$$

where

$$\mathbf{J} = \frac{\partial H_\epsilon}{\partial\mathbf{\Phi}}\frac{\partial\mathbf{\Phi}}{\partial\mathbf{x}}\frac{\partial\mathbf{W}}{\partial\Delta\mathbf{p}} = \delta_\epsilon(\mathbf{\Phi}(\mathbf{x}_i))\nabla\mathbf{\Phi}(\mathbf{x}_i)\frac{\partial\mathbf{W}}{\partial\Delta\mathbf{p}}. \tag{16}$$

$\delta_\epsilon(z)$ is the derivative of $H_\epsilon(z)$, i.e. a smoothed Dirac delta function. After substituting Eqs. (14) and (15) into Eq. (13), and setting the derivative of Eq. (13) to zero, we arrive at an expression for $\Delta\mathbf{p}$:

$$\Delta\mathbf{p} = \left[\sum_{n=1}^{N}\left(\frac{1}{2P(\mathbf{x}_n|\mathbf{\Phi},\mathbf{p},M)^{old}P(\mathbf{y}_n|M)^{old}}\left(\frac{P(\mathbf{y}_n|M_f)}{h} + \frac{P(\mathbf{y}_n|M_b)}{1-h}\right) + \frac{P(\mathbf{x}_n|F_v)}{2P(\mathbf{x}_n|F)^{old}\cdot h}\right)\mathbf{J}^T\mathbf{J}\right]^{-1} \cdot \sum_{n=1}^{N}\left(\frac{(P(\mathbf{y}_n|M_f) - P(\mathbf{y}_n|M_b))}{P(\mathbf{x}_n|\mathbf{\Phi},\mathbf{p},M)^{old}P(\mathbf{y}_n|M)^{old}} + \frac{P(\mathbf{x}_n|F_v)}{P(\mathbf{x}_n|F)^{old}}\right)\mathbf{J}^T. \tag{17}$$

where $P(\mathbf{x}_n|\mathbf{\Phi},\mathbf{p},M)^{old}, P(\mathbf{y}_n|M)^{old}$ and $P(\mathbf{x}_n|F)^{old}$ are obtained from previous frame, while $P(\mathbf{y}_n|M_f), P(\mathbf{y}_n|M_b)$ and $P(\mathbf{x}_n|F)$ are obtained from current frame. Eq. (17) is then used to update pose $\mathbf{p}$ by composing $\mathbf{W}(\mathbf{x},\mathbf{p})$ with $\mathbf{W}(\mathbf{x},\Delta\mathbf{p})^{-1}$.

Compared with the foreground probability map (e.g. Fig. 2(b)) produced by color histograms, voting map (e.g. Fig. 2(c)) provides extra spatial information to help find the position of the object robustly. This is especially important when background has similar colors with the object, and simple color histograms may have difficulty in distinguishing foreground pixels from background pixels. Hence, the foreground map and the voting map are complementary by nature, and the main goal of the rigid registration is to warp the new frame such that it matches better the old frame for both foreground/background map and voting map.

5. Segmentation

For segmentation, we optimize Eq. (9) with respect to the shape $\mathbf{\Phi}$, and keep the pose parameters $\mathbf{p}$ fixed. Instead of Hough voting map F_v, the set of support points F_s now serves as the spatial constraints on pixel location $\mathbf{x}_n$, which take the form:

$$P(\mathbf{x}_n|F_s) = \begin{cases} \exp(\beta H_\epsilon(\mathbf{\Phi}(\mathbf{x}_n)), & \text{if } \mathbf{x_n} \in F_s \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where β is the weight. This term encourages the support points to have positive level-set embedding function value $\mathbf{\Phi}(\mathbf{x}_n)$. In our level-set segmentation, $\mathbf{\Phi}(\mathbf{x}_n) > 0$ means that the pixel $\{\mathbf{x}_n, \mathbf{y}_n\}$ belong to foreground region. As the level-set embedding function is smooth in every pixel location, the support points will also influence the nearby pixels to become foreground pixels. In this way, the set of support points appears to be the foreground prior that guides the level-set evolution.

Substitute Eq. (18) into Eq. (9), and calculate the derivative of Eq. (9):

$$\frac{\partial P(\mathbf{\Phi},\mathbf{p}|\mathbf{x},\mathbf{y},F)}{\partial\mathbf{\Phi}} = \frac{\delta_\epsilon(\mathbf{\Phi})(P(\mathbf{y}_n|M_f) - P(\mathbf{y}_n|M_b))}{P(\mathbf{x}_n|\mathbf{\Phi},\mathbf{p},M)P(\mathbf{y}_n|M)} + \frac{\partial(\log(P(\mathbf{x}_n|F_s)))}{\partial\mathbf{\Phi}} - \frac{1}{\sigma^2}\left[\nabla^2(\mathbf{\Phi}) - \mathrm{div}\left(\frac{\nabla\mathbf{\Phi}}{|\nabla\mathbf{\Phi}|}\right)\right] - \lambda\delta_\epsilon(\mathbf{\Phi})\mathrm{div}\left(\frac{\nabla\mathbf{\Phi}}{|\nabla\mathbf{\Phi}|}\right) \tag{19}$$

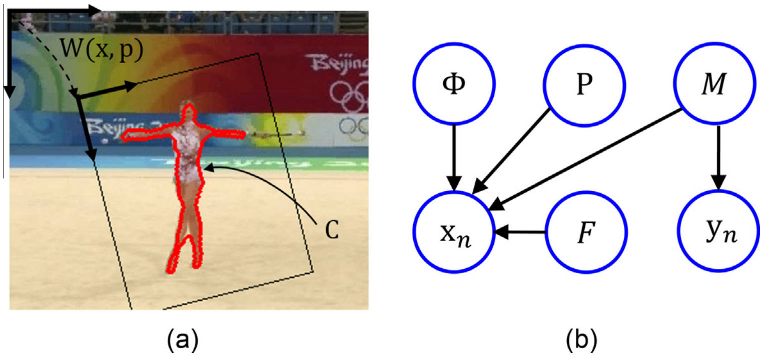

Fig. 3. (a) The warp $\mathbf{W}(\mathbf{x},\mathbf{p})$ describes the position of the object in the image, while the contour $\mathbf{C}$ separates the object from the background. (b) Graphical representation of the generative model used in our method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where

$$\frac{\partial(\log(P(\mathbf{x}_n|F_s)))}{\partial\mathbf{\Phi}} = \begin{cases}\beta\delta_\epsilon(\mathbf{\Phi}(\mathbf{x}_n)), & \text{if } \mathbf{x_n}\in F_s\\ 0, & \text{otherwise}\end{cases} \tag{20}$$

∇^2 is the Laplacian operator. Through gradient descent optimization method, we can find the optimum $\mathbf{\Phi}$. For some videos that contain complex scenes, our segmentation results will include some small isolated areas. We suggest to apply a pass of Gaussian filtering as a post-processing step to exclude these isolated areas and make the contour of the object smoother.

6. Models updating

After tracking and segmentation in each frame, we update both the Hough Forests model and the color histograms model with the segmentation results. To update Hough Forests, the image patches in object region are treated as positive samples, and the image patches from a rectangular frame surrounding the object's bounding box are treated as negative samples. To update color histograms, the color distribution of the support points is used to update foreground model M_f in a linear-opinion-pools manner, since these points are more reliable. The background model M_b is updated from the same region that provides negative samples for updating Hough Forests.

7. Experiments

The initialization of our method is a bounding box around the object in the first frame, given by the user. The level-set embedding function $\mathbf{\Phi}$ is initialized as a signed distance function according to the bounding box. Then $\mathbf{\Phi}$ is evolved according to Eq. (14), without using the support points constraints. At each iteration,the foreground and background histograms are rebuilt with the updated contour. In this way, we obtain a fine contour and accurate color histograms in the first frame. Based on this contour, the Hough Forests model is initialized.

In our implementation, similar to [8], the Hough Forests model consists 20 trees and maximum depth of each tree is 8. The patch size of the samples used in Hough Forests is 12 × 12. As suggested by [1,4], the weight λ, σ in Eq. (8) are set to 2 and $\sqrt{50}$ respectively. The weight β that controls the impact of support points in Eq. (13) is set to 5 through experiments. The parameter ϵ in Heaviside step function and Diarc delta function is set to 10. The maximum iteration number of solving Eq. (12) for rigid registration is set to 50, while the maximum iteration number of level-set evolution is set to 30.

We measured the average processing speed of our method for the all testing sequences on an Intel Core i3 CPU@ 2.93 GHz (we used only one core). Our method runs about 1.6 frames per second with the C++ implementation. The rigid registration accounts for about 70% of the total runtime consumption, while level-set evolution accounts for about 20%. Hough Forests is a more complex appearance model than color histograms, and Hough voting, back-projection and updating are all relatively time-consuming process. There is still room to speed up our method, such as using the level-set parallelization or a more efficient implementation of Hough Forests. We make quantitative comparison with several state-of-the-art trackers on two benchmarks.

7.1. Non-rigid object tracking dataset

We first conduct quantitative evaluation of our method on the non-rigid object tracking dataset, which is commonly used in previous works of non-rigid object tracking [8,14]. This dataset is composed of 11 challenging sequences, and has about 2500 frames in total. All sequences show moving objects under considerable rigid and non-rigid deformations. We compare our method with several popular object tracking methods: Struck [21], a bounding-box-based object tracking method; PWP [1], a level-set tracking method that relies on color histograms to distinguish object from background; HoughTrack [8], a segmentation-based tracking method that combines online Hough Forests tracking with GrabCut segmentation. For Struck and HoughTrack, we use the code provided by the original authors. For PWP, we implemented it by ourselves. Although our method can produce the segmentation

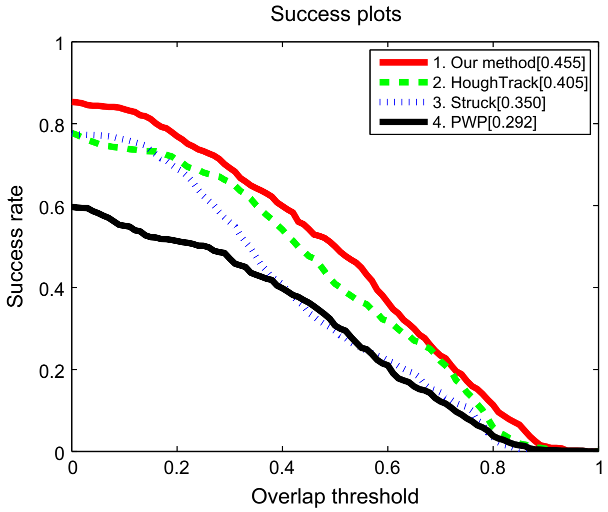

Fig. 4. The success plot showing rates of successful tracked frames at the thresholds varied from 0 to 1. All compared methods are ranked based on the performance score, as shown in the legend.

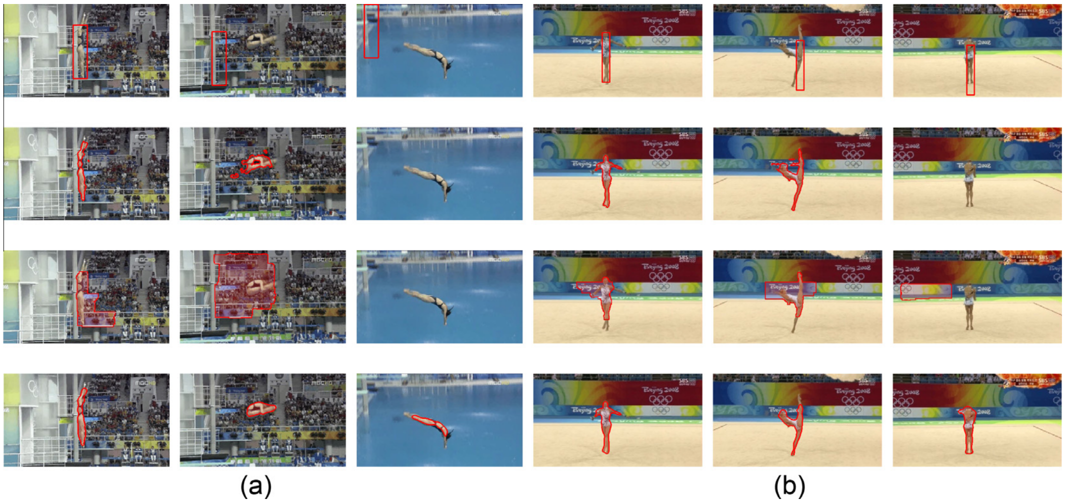

Fig. 5. Tracking results of different methods for (a) the "Diving" sequence and (b) the "Gymnastics" sequence in the non-rigid object tracking dataset. The first row: Struck [21], the second row: PWP [1], the third row: HoughTrack [8], the bottom row: our method. An image without any tracking mark means the method loses the track there.

results, we convert our results to the bounding boxes that cover the segmentation. And we do the same with results of PWP and HoughTrack, such that all the four methods can be compared in a standard form. To measure the performance of different tracking methods, we use the percentage of successfully tracked frames. Let BB_T denote the bounding box produced by tracking, and BB_G denote the ground-truth bounding box. In each frame, the object is considered as successfully tracked if the overlap measure $\frac{BB_T \cap BB_G}{BB_T \cup BB_G}$ is above a threshold.

Instead of using a specific threshold for evaluation, we vary the threshold from 0 to 1 and calculate the rates of successfully tracked frames at each threshold. The success plot is shown in Fig. 4. From the plot, our method outperforms other compared tracking methods at all overlap thresholds. Fig. 5 illustrates some tracking results of the four compared methods for the "Diving" and "Gymnastics" sequence. Struck uses a bounding box representation, which is not the appropriate way to describe articulated object. PWP utilizes a simple color histogram model, and its ability to capture rapid appearance changes is relatively limited. HoughTrack relies on the Grabcut algorithm to segment the object, which may have some problems in very cluttered background. Our method can track and segment the object successfully in the both sequences. And more comparisons can be found in Supplementary material.

7.2. CVPR2013 tracking benchmark

The second experiment is conducted on the CVPR2013 online object tracking benchmark [10]. This benchmark is mainly designed for evaluation of bounding-box-based tracking methods which perform only tracking without segmentation. Although the goal of our method is to track and segment object simultaneously, we still evaluate our method on this benchmark and compare with the state-of-the-arts in the tracking performance.

The whole dataset consists of 50 fully annotated sequences, most of which are widely used in the online tracking literature over the past several years. For better evaluation and analysis of the strength and weakness of different methods, each sequence is tagged with a number of attributes indicating to the presence of different challenges, such as occlusion, fast motion and background clutters. According to these attributes, all the sequences are categorized and therefore 11 challenge subsets are created. The benchmark use precision plot and success plot as two different evaluation metrics. The precision plot shows the percentage of frames on which the center location error is within the a given threshold, while the success plot shows percentage of frames whose bounding box overlap score (Section 7.1) is above a threshold. Both location error threshold and overlap threshold are varied

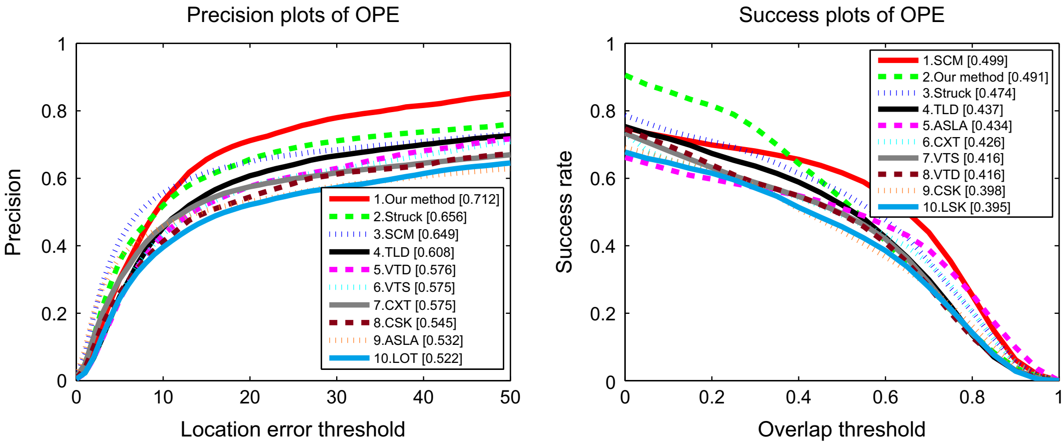

Fig. 6. Quantitative comparison in CVPR2013 benchmark. The performance score of each tracking method is shown in the legend. In each figure, the top 10 tracking methods are presented, in the order of performance score. The tracking methods appearing in the legend are as follows: Struck [21], SCM [22], TLD [23], VTD [24], VTS [25], CXT [26], CSK [27], ALSA [28], LOT [29], LSK [30].

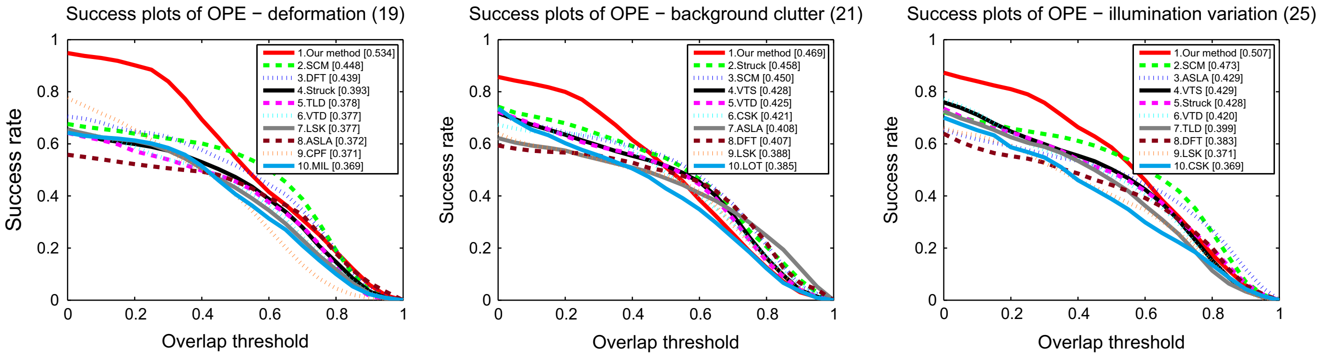

Fig. 7. Success plots for some challenge subsets of CVPR2013 tracking benchmark. In each figure, the value appears in the title indicates the number of sequences in that subset, and the top 10 tracking methods are presented, in the order of tracking performance score. The tracking methods appearing in the legend are as follows (besides the tracking methods presented in Fig. 6): DFT [31], CPF [32], MIL [33].

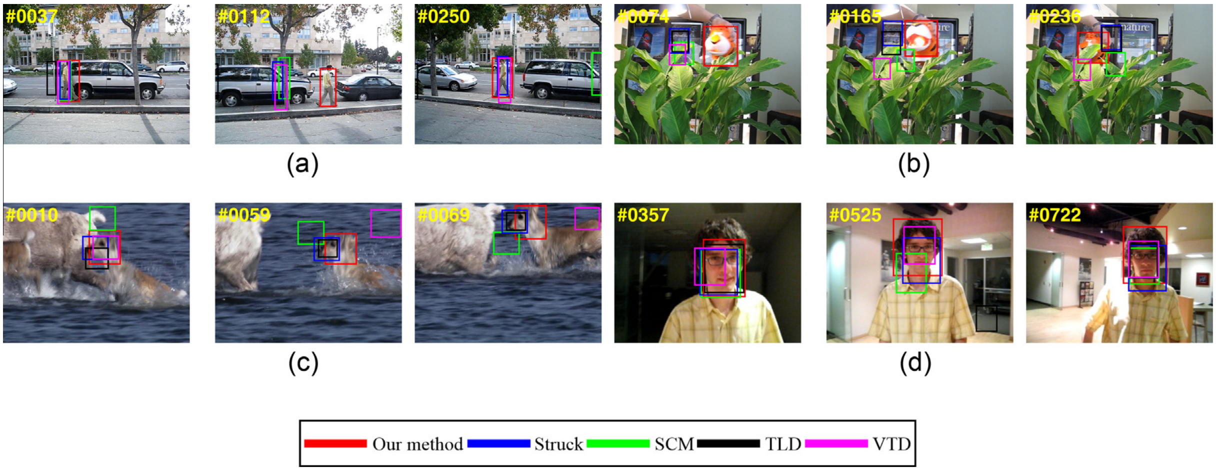

Fig. 8. Some tracking results on four sequences in the CVPR2013 benchmark obtained by the top five tracking methods (Our method, Struck [21], SCM [22], TLD [23] and VTD [24]) in the precision plot. (a) “David3” sequence. (b) “Tiger1” sequence. (c) “Deer” sequence. (d) “David” sequences.

for a comprehensive comparison. The benchmark currently includes 29 popular tracking methods, and provide a code library with uniform input and output. We refer readers to the original paper [10] for more details.

We run the One-Pass Evaluation (OPE) [10] on the benchmark using our method, and convert our segmentation results to the bounding box representation. To make comparison, we directly use the online available tracking results of the 29 tracking methods on this benchmark. The overall performance of OPE for our method and other state-of-the-arts (ranked within top 10) are shown in Fig. 6. In the precision plot, our proposed method outperforms all other 29 tracking methods. In the success plot, our method ranks the second, slightly falling behind SCM [22]. As shown in Fig. 7, our tracking method is more robust to deformation, background clutter and illumination variation comparing with other tracking methods. Especially in the deformation subset, our method outperforms the second best SCM [24] by nearly 20%, which demonstrates that segmentation is beneficial for handling topological changes cased by non-rigid object deformation. For more intuitive demonstration, qualitative comparison with top-rank tracking methods is shown in Fig. 8.

8. Conclusions

In this paper, we present a level-set probabilistic framework for tracking non-rigid object in a video sequence without any prior knowledge. We make use of the two features of the Hough Forests model: Hough voting and support points, which serve as the spatial constraints for tracking and segmentation. In each frame, we solve rigid registration and level-set evolution by optimizing the derived energy functions, and finally update the appearance models according to the segmentation result. We conduct experiments on two tracking benchmark datasets, and compare with other state-of-the-art tracking methods comprehensively. Experimental results demonstrate that our tracking method can perform comparably to the state-of-the-arts in the CVPR13 tracking benchmark, while shows significantly improved performance in non-rigid object tracking aspect.

In future work, we plan to apply our method in some special scenarios, such as hand tracking or pedestrian tracking. Since we track a special class of objects, some priors can be added to enhance the tracking performance. On the other hand, tracking with RGBD data is becoming more and more popular. Although the depth data can provide some reliable information with illumination invariance to help tracking, how to integrate the advantages of RGB data and depth data is a interesting direction. Considering that currently our method is a little slow, we think a implementation on GPU would be beneficial.

Acknowledgment

We thank the editor and the anonymous reviewers for their constructive comments. This work was partially supported by NSFC (Nos. 61379068 and 61402387), and the Open Project Program of State Key Lab of CAD&CG, Zhejiang University (No. A1419).

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.jvcir.2016.04.009.

References

[1] C. Bibby, I.D. Reid, Robust real-time visual tracking using pixel-wise posteriors, in: Proc. ECCV, 2008.

[2] C. Bibby, I.D. Reid, Real-time tracking of multiple occluding objects using level sets, in: Proc. CVPR, 2010.

[3] D. Mitzel, E. Horbert, A. Ess, B. Leibe, Multi-person tracking with sparse detection and continuous segmentation, in: Proc. ECCV, 2010.

[4] E. Horbert, K. Rematas, B. Leibe, Level-set person segmentation and tracking with multi-region appearance models and top-down shape information, in: Proc. ICCV, 2011.

[5] V.A. Prisacariu, I.D. Reid, PWP3D: real-time segmentation and tracking of 3D objects, Int. J. Comput. Vis. 98 (3) (2012) 335–354.

[6] P.F. Felzenszwalb, R.B. Girshick, D.A. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.

[7] J. Gall, A. Yao, N. Razavi, L.J.V. Gool, V.S. Lempitsky, Hough forests for object detection, tracking, and action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 33 (11) (2011) 2188–2202.

[8] M. Godec, P.M. Roth, H. Bischof, Hough-based tracking of non-rigid objects, in: Proc. ICCV, 2011.

[9] A.W.M. Smeulders, D.M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, Visual tracking: an experimental survey, IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2014) 1442–1468.

[10] Y. Wu, J. Lim, M.-H. Yang, Online object tracking: a benchmark, in: Proc. CVPR, 2013, pp. 2411–2418.

[11] S.M. Shahed Nejhum, J. Ho, M.-H. Yang, Online visual tracking with histograms and articulating blocks, Comput. Vis. Image Understand. 114 (8) (2010) 901–914.

[12] J. Fan, X. Shen, Y. Wu, Scribble tracker: a matting-based approach for robust tracking, IEEE Trans. Pattern Anal. Mach. Intell. 34 (8) (2012) 1633–1644.

[13] V. Belagiannis, F. Schubert, N. Navab, S. Ilic, Segmentation based particle filtering for real-time 2d object tracking, in: Proc. ECCV, 2012.

[14] S. Duffner, C. Garcia, Pixeltrack: a fast adaptive algorithm for tracking non-rigid objects, in: Proc. ICCV, 2013.

[15] D. Cremers, Dynamical statistical shape priors for level set-based tracking, IEEE Trans. Pattern Anal. Mach. Intell. 28 (8) (2006) 1262–1273.

[16] X. Sun, H. Yao, S. Zhang, A novel supervised level set method for non-rigid object tracking, in: Proc. CVPR, 2011.

[17] R.B. Girshick, J. Shotton, P. Kohli, A. Criminisi, A.W. Fitzgibbon, Efficient regression of general-activity human poses from depth images, in: Proc. ICCV, 2011.

[18] M. Dantone, J. Gall, G. Fanelli, L.J.V. Gool, Real-time facial feature detection using conditional regression forests, in: Proc. CVPR, 2012.

[19] B. Leibe, A. Leonardis, B. Schiele, Robust object detection with interleaved categorization and segmentation, Int. J. Comput. Vis. 77 (1–3) (2008) 259–289.

[20] S. Baker, I. Matthews, Lucas-kanade 20 years on: a unifying framework, Int. J. Comput. Vis. 56 (3) (2004) 221–255.

[21] S. Hare, A. Saffari, P.H.S. Torr, Struck: structured output tracking with kernels, in: Proc. ICCV, 2011.

[22] W. Zhong, H. Lu, M.-H. Yang, Robust object tracking via sparsity-based collaborative model, in: Proc. CVPR, 2012, pp. 1838–1845.

[23] Z. Kalal, J. Matas, K. Mikolajczyk, P-n learning: bootstrapping binary classifiers by structural constraints, in: Proc. CVPR, 2010.

[24] J. Kwon, K.M. Lee, Visual tracking decomposition, in: Proc. CVPR, 2010, pp. 1269–1276.

[25] J. Kwon, K.M. Lee, Tracking by sampling trackers, in: Proc. ICCV, 2011, pp. 1195–1202.

[26] T.B. Dinh, N. Vo, G.G. Medioni, Context tracker: Exploring supporters and distracters in unconstrained environments, in: Proc. CVPR, 2011, pp. 1177–1184.

[27] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, Exploiting the circulant structure of tracking-by-detection with kernels, in: Proc. ECCV, 2012, pp. 702–715.

[28] X. Jia, H. Lu, M.-H. Yang, Visual tracking via adaptive structural local sparse appearance model, in: Proc. CVPR, 2012, pp. 1822–1829.

[29] S. Oron, A. Bar-Hillel, D. Levi, S. Avidan, Locally orderless tracking, in: Proc. CVPR, 2012, pp. 1940–1947.

[30] B. Liu, J. Huang, L. Yang, C.A. Kulikowski, Robust tracking using local sparse appearance model and k-selection, in: Proc. CVPR, 2011, pp. 1313–1320.

[31] L. Sevilla-Lara, E.G. Learned-Miller, Distribution fields for tracking, in: CVPR, 2012, pp. 1910–1917.

[32] P. Pérez, C. Hue, J. Vermaak, M. Gangnet, Color-based probabilistic tracking, in: ECCV, 2002, pp. 661–675.

[33] B. Babenko, M. Yang, S. Belongie, Robust object tracking with online multiple instance learning, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1619–1632.