# Level Set Based Online Visual Tracking via Convolutional Neural Network

Xiaodong Ning and Lixiong Liu(✉)

Beijing Laboratory of Intelligent Information Technology,
School of Computer Science and Technology,
Beijing Institute of Technology, Beijing 100081, China
{xdning, lxliu}@bit.edu.com

**Abstract.** In this paper, we propose a level set tracking algorithm, which integrates the information of the original frame and the confidence predicted by the deep feature based detector. First, we extract features from convolutional neural network and select part of them to avoid redundancy. Secondly, the features are used to generate a confidence map of the tracked object through the detector. And then the confidence along with the original frame is applied in level set model to acquire the segmentation result. We introduce an outlier rejection scheme to further improve the result. Finally, updating is employed to the detector to adapt to the changes in the video. One important contribution of our work is to use the deep features in confidence prediction, particularly the usage of low-level features in the neural network. Experimental results show that our model delivers a better performance than the state-of-the-art on a series of challenging videos.

**Keywords:** Object tracking · Level set · Convolutional neural network · Deep feature

## 1 Introduction

Object tracking is a fundamental task in the area of computer vision. It has a large variety of subfields, e.g. location tracking, trajectory tracking and contour tracking, which are classified by the characteristics or properties it tracks [1]. We are focused on contour tracking in this paper. A typical contour tracking algorithm aims to segment the object in the following each frame with an initialization in the first one. It is a challenging problem due to the complicated appearance changes in the videos.

Prior researches [2–4] mostly depend on hand-crafted features to build segmentation models. Those traditional features are limited in recognizing the sudden change of the object and not robust to complex scenes. Differing from conventional methods, deep feature based tracking algorithms [5, 6] demonstrate a remarkable performance. Deep features are extracted from convolutional neural networks (CNNs) and have strong capabilities of distinguishing objects.

While the performance is promising, limitations still remain, e.g. the deep features at a relatively low level are often ignored. Most current methods only adapt higher layer features such as the fully-connected layer features in R-CNN network [7],


D. Liu et al. (Eds.): ICONIP 2017, Part III, LNCS 10636, pp. 280–290, 2017.
https://doi.org/10.1007/978-3-319-70090-8_29

conv4-3 and conv5-3 layers in VGG network [8] and conv3 to conv5 in VGG [9]. Consequently, the intra class variations encoded by lower layers are discarded, which prevents accurate segmentation of the object contour. Therefore, our model includes some low-level features, and deploys an AdaBoost selection model [10] to remove unrelated feature maps. Hence we can both select best features and predict a confidence map efficiently. An outlier rejection scheme and a different updating scheme from [10] are introduced to further improve the effectiveness of our model.

Besides, current methods [7, 11] apply graph-cut segmentation and this leads to relatively slow efficiency [12]. Some attempts have been made to reduce the computational cost such as correlation filter [9] and probabilistic soft segmentation [13], whereas the segmentation of the object is not satisfactory. Instead, we develop a novel level set method. Level set methods are widely used in image segmentation problems, e.g. Chan-Vese model [14] and Liu et al.'s method [15], and show high efficiency in tracking problem [16]. Meanwhile, present level set models lack prior knowledge restrictions, which lead to unnecessary topological changes. Thus we construct a level set energy function which contains new length and weighted area terms. It balances the result of appearance model and the image information, and ensures the level set evolution can be restricted by the appearance model result.

In this paper, we propose an online visual tracking algorithm which consists of a deep feature based discriminative model and a level set method. The contributions of our work are as follows. First, we build a framework using a deep feature based model and level set method, where the CNN based detector can be updated online. Second, we improve the discriminative model which uses both low-level and relatively high-level deep features to detect the location of the object and obtains a coarse map for segmentation. Finally, we incorporate the result of detector with information of the original image in the energy function of the level set method, yielding a novel segmentation model.

## 2 Proposed Algorithm

The proposed tracker mainly contains two components: an appearance model to generate the confidence map of the object (i.e. an object detector) and a segmentation model to acquire the contour. An overview of our algorithm is shown in Fig. 1. For a given initial contour and its corresponding frame, a series of feature maps are first extracted from the region of interest (ROI) of the frame. The features include both hand-crafted features and deep features from VGG network. Then an AdaBoost feature selection process is performed on the extracted feature maps to reduce redundant ones. A discriminative model is also learned through the features and the contour in this procedure. For each subsequent frame, a ROI centered at the object contour in the last frame is cropped and the selected features are extracted. Finally, the confidence map is predicted by the discriminative model and a level set algorithm based on confidence is used to generate the object contour.

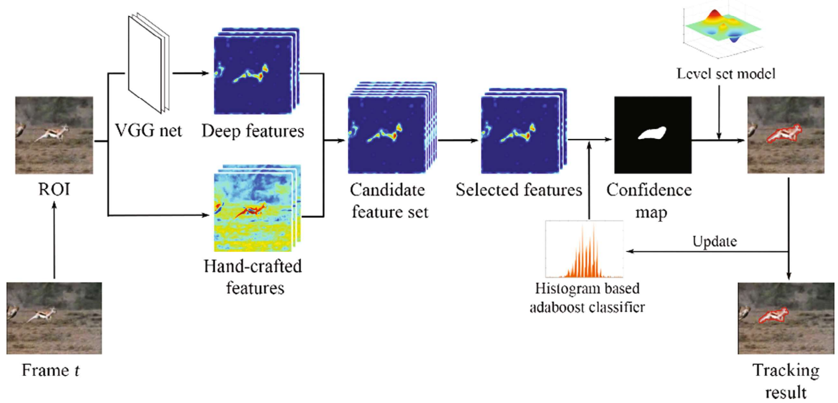



**Fig. 1.** Pipeline of the proposed algorithm. For Frame $t$, our tracker first extracts the deep features and handcrafted features. Then the features are selected according to the arrangement at the initialization stage of the tracking. Confidence map is predicted by using the selected features and an AdaBoost model. Finally, we use a level set method based on confidence to segment Frame $t$. Weak classifiers in the AdaBoost model are updated online while the arrangement of features is initialized at the beginning of tracking and will not be changed.

### 2.1 AdaBoost Feature Selection and Model Training

Our selection model is based on [10], an AdaBoost feature selection method. An extra outlier rejection scheme and updating are also introduced (see Sects. 2.3 and 2.4). This model can be used in feature selection at the initialization stage of our algorithm and confidence prediction during tracking.

In our model, Candidate feature set F contains hand-crafted features and VGG16 net [17] features. The hand-crafted features are extracted in both RGB and HSV color space in order to prevent the insufficiency of color information caused by a single space. Meanwhile, deep features including all the convolutional layers in VGG are shown in Fig. 2. In CNNs, the resolution of maps reduces along with the increasing of layer depth due to pooling operators. So we abandon higher layer features because they encode too small contour information of the object.

A detailed review of the extracted 1486-dim candidate features is shown in Table 1.

**Table 1.** Extracted candidate feature set.

| Feature ID | Feature description | Feature ID | Feature description |
|---|---|---|---|
| $f_1$–$f_3$ | RGB color | $f_{79}$–$f_{206}$ | conv2-2 layer features |
| $f_4$–$f_6$ | HSV color | $f_{207}$–$f_{462}$ | conv3-3 layer features |
| $f_7$–$f_{14}$ | 8-oriented HOG descriptors calculated on a 5 × 5 window | $f_{463}$–$f_{974}$ | conv4-3 layer features |
| $f_{15}$–$f_{78}$ | conv1-2 layer features | $f_{975}$–$f_{1486}$ | conv5-3 layer features |

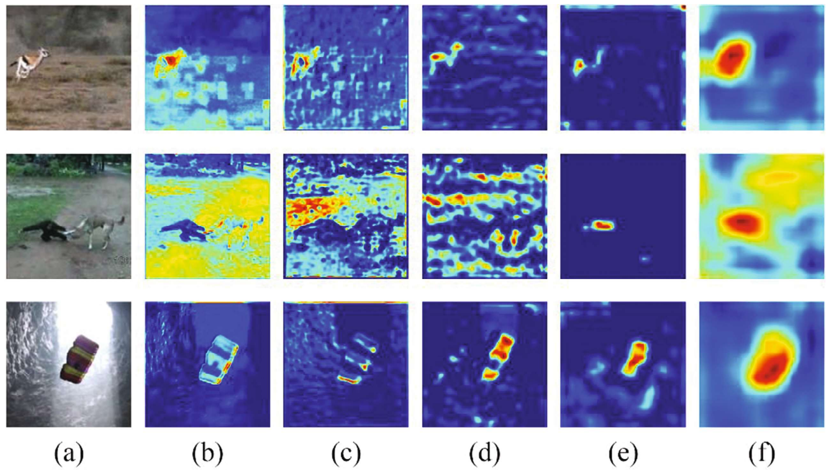



**Fig. 2.** Visualization of feature maps from different VGG convolutional layers. (a) ROIs of original frames. (b)–(f) features extracted from conv1-2, conv2-2, conv3-3, conv4-3 and conv5-3 layers respectively. It can be seen that the latter layer features have strengths at semantical discrimination of the object. However, the details of the object contour suffer loss with the network forwarding.

Given a single feature map $f \in \mathrm{F}$, the weak classifier is defined as:

$$\mathrm{L}(f) = \log \frac{\max\{\mathrm{H}_{\mathrm{Obj}}(f), \delta\}}{\max\{\mathrm{H}_{\mathrm{Bg}}(f), \delta\}} \tag{1}$$

where $\mathrm{H}_{\mathrm{Obj}}$ and $\mathrm{H}_{\mathrm{Bg}}$ are the histograms of object and background, and $\delta$ is a small value fixed to 0.001 that prevents dividing by zero. Feature map $f$ is also tuned to have better discrimination by Eq. (1) [18].

The error of each feature is calculated with the weak classifier result $\mathrm{LI}_f = \mathrm{L}(f)$ and the sample weight $\mathrm{w}_\mathrm{i}^0$. Initially, the sample weight $\mathrm{w}_\mathrm{i}^0 = 1/\mathrm{N}$ is equal for each feature where N is the number of samples. And at the $t$th iteration, error of the feature $f_t$ is weighted by $\mathrm{W}^t = \{\mathrm{w}_\mathrm{i}^t | 1 \leq \mathrm{i} \leq \mathrm{N}\}$:

$$\mathrm{Error}(\mathrm{W}^t; f_t) = \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{w}_\mathrm{i}^t \left(\mathrm{sgn}\left(\mathrm{LI}_{f_t}(\mathrm{i})\right) \neq \mathrm{y}_\mathrm{i}\right) \tag{2}$$

$\mathrm{y}_\mathrm{i} \in \{1, -1\}$ is the training label which represents the sample belong to the object ($\mathrm{y}_\mathrm{i} = 1$) or background ($\mathrm{y}_\mathrm{i} = -1$).

The weak classifier weight $\alpha_t$ for the feature $f_t$ is given as follows:

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \mathrm{Error}(\mathrm{W}^t; f_t)}{\mathrm{Error}(\mathrm{W}^t; f_t)}\right) \tag{3}$$

At every iteration, we update $W^t$ according to the error of weak classifier $LI_{f_t}$,

$$w_i^{t+1} = w_i^t \exp\left(-\alpha_t \mathrm{sgn}\left(LI_{f_t}(i) \neq y_i\right)\right) \tag{4}$$

The feature maps with T minimal errors are selected for predicting confidence map S. Thus, the confidence is

$$S = \mathrm{sgn}\left(\sum\nolimits_{t=0}^{T-1} \alpha_t LI_{f_t}\right) \tag{5}$$

It is worth noting that T is much less than the size of candidate feature set, because we can eliminate noise in confidence map and get a high predicting efficiency by a small T.

### 2.2 Level Set Model

We develop a level set method to further segment the object in the frame. The initial level set $\phi_t$ is determined by the segmentation result $\phi_{t-1}$ of the last frame ($\phi$ is defined by the initial contour $C_0$ for the first frame),

$$\phi_t(x) = \begin{cases} -1, & \text{if } \phi_{t-1}(x) \circ G_d < 0 \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

where $\phi_{t-1} \circ G_d$ is a dilation operation with an element $G_d$ adapted on the $\phi_{t-1}$.

Like the traditional level set methods [16], we also used the edge indicator in our model. Let I be the frame represented in gray value, the conventional edge indicator can be defined as

$$g = \frac{1}{1 + |\nabla G_\sigma * I|^2} \tag{7}$$

where $G_\sigma$ is Gaussian kernel with a standard deviation $\sigma$. This edge indicator is employed in the length and area terms respectively,

$$\mathcal{A}(\phi) = \int_\Omega g H_\varepsilon(-\phi) dx \tag{8}$$

$$\mathcal{L}(\phi) = \int_\Omega g \delta_\varepsilon(\phi) |\nabla \phi| dx \tag{9}$$

where $\Omega$ is the whole image domain, and $H_\varepsilon$ is Heaviside function with parameter $\varepsilon$,

$$H_\varepsilon(x) = \begin{cases} \frac{1}{2}\left(1 + \frac{x}{\varepsilon} + \frac{1}{\pi}\sin\left(\frac{\pi x}{\varepsilon}\right)\right), & |x| \leq \varepsilon \\ 1, & x > \varepsilon \\ 0, & x < -\varepsilon \end{cases} \tag{10}$$

The area term $\mathcal{A}(\phi)$ computes a weighted area the region inside the contour, while $\mathcal{L}(\phi)$ computes the energy along the length of the contour. In our model, the novel area and length terms adopt the confidence map after erosion and dilation operation $S_d = S \circ G_e \circ G_d$ to guide the edge indicator in both the area term and length term,

$$\mathcal{A}'(\phi) = \int_{\Omega} g_{\mathcal{A}} H_{\varepsilon}(-\phi)dx, \quad g_{\mathcal{A}} = \frac{1}{2}\left(\frac{1 - S_d}{2} + g\right) \tag{11}$$

$$\mathcal{L}'(\phi) = \int_{\Omega} g_{\mathcal{L}} \delta_{\varepsilon}(\phi)|\nabla\phi|dx, \quad g_{\mathcal{L}} = \frac{1}{2}\left(\frac{1}{1 + |\nabla S_d|^2} + g\right) \tag{12}$$

where $\frac{1-S_d}{2} \in \{0, 1\}$ can speed up the evolution process toward the predicted map S, and $\frac{1}{1+|\nabla S_d|^2}$ is boundary adjustment to the edge indicator g. The proposed indicators use the linear combination to compensate for the errors caused by g, and avoid the segmentation failure when S is inaccurate.

Hence the energy function $E(\phi)$ is obtained by three energy terms, improved area term and length term $\mathcal{A}'(\phi)$, $\mathcal{L}'(\phi)$, and penalty term $\mathcal{P}(\phi)$ respectively,

$$E(\phi) = \alpha\mathcal{A}'(\phi) + \nu\mathcal{L}'(\phi) + \mu\mathcal{P}(\phi) \tag{13}$$

where $\mathcal{P}(\phi)$ is

$$\mathcal{P}(\phi) = \frac{1}{2}\int_{\Omega} (|\nabla\phi| - 1)^2 dx \tag{14}$$

where $\delta_{\varepsilon} = H'_{\varepsilon}$ is the Dirac function,

$$\delta_{\varepsilon}(x) = \begin{cases} \frac{1}{2\varepsilon}\left(1 + \cos\left(\frac{\pi x}{\varepsilon}\right)\right), & |x| \leq \varepsilon \\ 0, & |x| > \varepsilon \end{cases} \tag{15}$$

We can use the following gradient flow to minimize the energy function Eq. (13),

$$\frac{\partial\phi}{\partial t} = \mu\left(\Delta\phi - \mathrm{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)\right) + \nu\delta_{\varepsilon}(\phi)\mathrm{div}\left(g_{\mathcal{L}}\frac{\nabla\phi}{|\nabla\phi|}\right) + \alpha g_{\mathcal{A}}\delta_{\varepsilon}(\phi) \tag{16}$$

The curve of zero level set can iteratively segment the object contour by minimizing $E(\phi)$ according to Eq. (16). Finally, we obtain the object contour $-\mathrm{sgn}(\phi)$.

### 2.3 Outlier Rejection

Since AdaBoost is sensitive to outliers [19], we introduce an outlier rejection scheme to compensate the discriminative model for wrong classification. Basically, the classification result after AdaBoost and segmentation can be written as

$$y_i(x) = \begin{cases} 1, & \text{if } \phi_t(x) < 0 \\ -1, & \text{otherwise} \end{cases} \quad (17)$$

The outlier rejection is defined as follows,

$$y'_i(x) = \begin{cases} 1, & \text{if } \phi_t(x) < 0 \wedge w_i < \Theta \wedge \|x - \text{center}(\phi_{t-1})\|_2 < D \\ -1, & \text{otherwise} \end{cases} \quad (18)$$

In Eq. (18), $w_i$ is sample weight for the pixel x. The weight for each pixel is first initialized equally as $w_i = 1/N$. Then the selected feature $f_t$ and the $y_i$ in Eq. (17) are used to compute the error and $\alpha_t$ according to Eqs. (2) and (3). We update $w_i$ by the error and $\alpha_t$ same as the Eq. (4). This scheme ensures the sample with a too large weight (larger than the predefined threshold $\Theta = 9/N$) is labeled negative. The samples that are too difficult to classify will be defined as background, which can lead to a cleaner result.

Moreover, we introduce a distance based rejection scheme. $\|\cdot\|_2$ means 2-norm, D is a threshold and $\text{center}(\phi_{t-1})$ is the center location of the object in the last frame in Eq. (18). The positive samples which are too far away from the object in the last frame will be removed because the object location in a video is usually continuous.

### 2.4 Model Updating

Once the segmentation and the outlier rejection are finished, the weak classifiers of AdaBoost model will be updated online. In Eq. (1), a weak classifier consists of two parts, object and background histograms. Thus we update the two histograms by linear combination of the current histograms and the ones in last frame,

$$P_t(c|H_i) = (1 - \gamma_i)P_{t-1}(c|H_i) + \gamma_i P_t(c|H_i), i = \{\text{Obj}, \text{Bg}\} \quad (19)$$

$\gamma_{Obj}$ and $\gamma_{Bg}$ are set to 0.08 and 0.1 respectively in all our experiments.

## 3 Experiments

Our model is implemented in C++ based on Caffee and OpenCV 2.4.10. All the tests run on a computer with a 3.3 GHz CPU, 4 GB RAM and a GK208 GPU. We select T = 6 best features to predict the confidence map in our experiments. The parameter setting for level set segmentation are shown in Table 2,

**Table 2.** Level set parameter setting.

| $n$ | $\Delta t$ | $\varepsilon$ | $\alpha$ | $\nu$ | $\mu$ |
|---|---|---|---|---|---|
| 20 | 1.0 | 1.5 | 3.0 | 6.0 | 0.2 |

where $n$ is the iteration number of evolution, $\Delta t$ is the step of iteration, and $\varepsilon$, $\alpha$, $\nu$ and $\mu$ are the same as parameters in Eqs. (10) and (13).

We evaluate our algorithm on SegTrack database [20] and compare it with other three top-performance non-rigid contour trackers [11, 13, 16]. The dataset consists of six sequences: birdfall2, cheetah, girl, monkeydog, parachute and penguin. Our tracker is evaluated by three metrics: success/precision plots of one-pass evaluation (OPE), error pixels and frame per second.

We first compare the success and precision plots of different trackers. For fair comparison, we convert the tracking contours of HT [11], SLSM [16] and our model into bounding boxes while the result of PT [13] is directly bounding box. The converted trackers are evaluated by success ratio, the ratio of frames whose tracked box has more overlap with ground truth box than the threshold; and precision, the ratio of frames whose tracking result is within the threshold from ground truth. Figure 3 shows the results of trackers. It is demonstrated that our model has better performance than other three trackers in both success and precision.

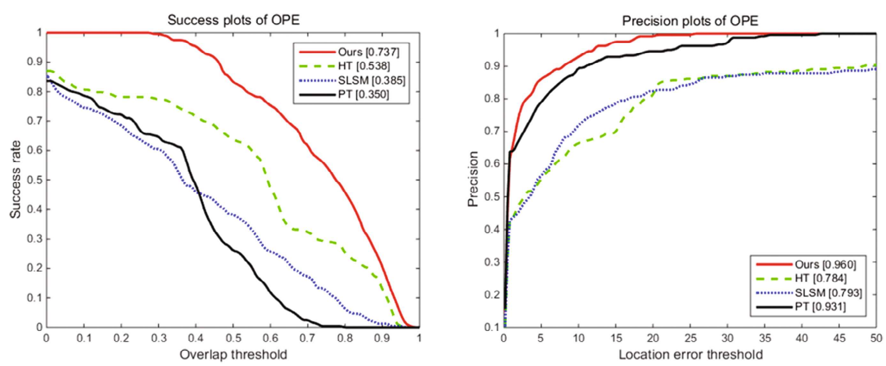


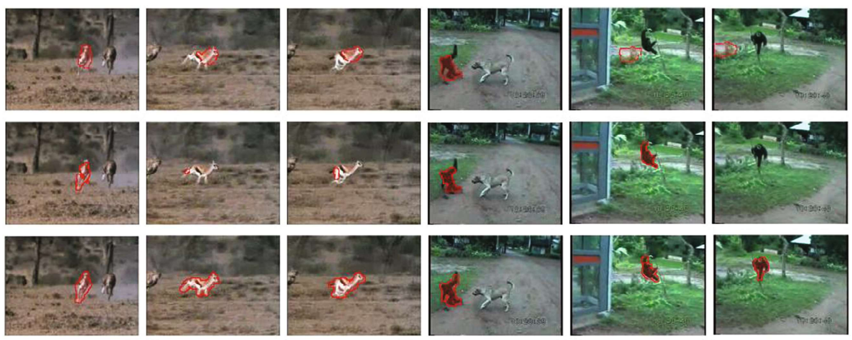

**Fig. 3.** The success plots and precision plots comparison of different trackers. The performance score of each tracker is presented in the legend. Values in the brackets are AUC (area under curve) and show the overall performance of trackers.

For the trackers whose results are in "contour" form, we further evaluate them by average number of error pixels per frame. As we discussed above, results of PT is in the form of bounding box and the metrics of error pixels is not fair for PT. Therefore, PT is not involved in this part. Table 3 lists the quantitative results of HT [11], SLSM [16] and our model. It can be seen that our model has smaller errors than HT and SLSM on the six videos. This advantage over SLSM, which also uses the level set method, may be due to more accurate confidence maps generated by deep features. Figure 4 shows the qualitative comparison in some selected frames from the dataset.

We measure the processing speed of all selected algorithms by frame per second in

**Table 3.** Average number of error pixels per frame of different trackers. The lowest errors are in boldface.

| Sequence | HT [11] | SLSM [16] | Our model |
|---|---|---|---|
| girl | 19995 | 10076 | **3462** |
| parachute | 587 | 2233 | **321** |
| penguin | 5597 | 5658 | **2918** |
| monkeydog | 1883 | 1080 | **560** |
| cheetah | 1193 | 1464 | **787** |
| birdfall2 | 327 | 430 | **261** |
| average | 4930 | 3490 | **1384** |



(a) (b)

**Fig. 4.** Tracking results of different models for (a) "cheetah" and (b) "monkeydog" sequences. The first row: HT [11], the second row: SLSM [16] and the bottom row: our method.

Table 4. PT is the fastest algorithm while SLSM is the second. Our model is slower than SLSM, but faster than HT. Moreover, our model obtains the similar FPS values on all the videos because it processes the frame within a $224 \times 224$ ROI and the computational complexity is almost the same for different sequences. This inferior time consumption to the level set based tracker, SLSM, is due to the fixed size of segmentation region. SLSM segments the object in a window which is twice the size of the object and this strategy usually generates a smaller ROI than ours. In future work, we plan to introduce acceleration scheme [21] to enhance the efficiency of our model. Besides, it should be noted that the speed of our model is acceptable, when it has far better performance than others and still can compete with high-performance algorithm, HT.

**Table 4.** Comparison of running speed of methods (evaluated by frame per second). The best results are in boldface.

| Sequence | HT [11] | PT [13] | SLSM [16] | Our model |
|---|---|---|---|---|
| girl | 0.1140 | **1.2494** | 0.4769 | 0.4920 |
| parachute | 0.2559 | **3.4570** | 2.0394 | 0.4927 |
| penguin | 0.1962 | **3.5360** | 1.1009 | 0.4957 |
| monkeydog | 0.3922 | **5.8909** | 3.1212 | 0.4941 |
| cheetah | 0.4317 | **6.2957** | 4.5045 | 0.4921 |
| birdfall2 | 0.6573 | 5.2128 | **9.9931** | 0.4885 |
| average | 0.3412 | **4.2736** | 3.5393 | 0.4925 |

## 4 Conclusion

We proposed a level set tracking framework which uses convolutional neural network features in this paper. We combine hand-crafted features and deep features, and then an AdaBoost selection procedure is employed on the features in order to select features and generate a coarse confidence map of the object. Afterwards, we segment the current frame based on the confidence along with the original image by a level set method. We also introduce an outlier rejection scheme and updating to further improve the result. Experimental results indicate that our method outperforms the state-of-the-art non-rigid contour tracking algorithms on SegTrack benchmark.

In future work, we will investigate how to improve the computational efficiency of our model, especially in the level set model. Other interests involve applying the superpixel to our model to generate more accurate confidence map, and exploring more effective segmentation methods.

**Acknowledgments.** This work is supported by National Natural Science Foundation of China under grant 61370133 and grant 61672095.

## References

1. Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., Hengel, A.V.D.: A survey of appearance models in visual object tracking. ACM Trans. Intell. Syst. Technol. **4**(4), 478–488 (2013)
2. Hu, W., Zhou, X., Li, W., Luo, W., Zhang, X., Maybank, S.: Active contour-based visual tracking by integrating colors, shapes and motions. IEEE Trans. Image Process. **22**(5), 1778–1792 (2013)
3. Bibby, C., Reid, I.: Robust real-time visual tracking using pixel-wise posteriors. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5303, pp. 831–844. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88688-4_61
4. Barbu, T.: Template matching based video tracking system using a novel n-step search algorithm and HOG features. In: International Conference on Neural Information Processing (ICONIP), Doha, pp. 328–336 (2012)
5. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, pp. 4293–4302 (2016)

6. Zhang, K., Liu, Q., Wu, Y., Yang, M.H.: Robust visual tracking via convolutional networks without training. IEEE Trans. Image Process. **25**(4), 1779–1792 (2016)
7. Hong, S., You, T., Kwak, S., Han, B.: Online tracking by learning discriminative saliency map with convolutional neural network. In: International Conference on Machine Learning, Lille, pp. 597–606 (2015)
8. Wang, L., Ouyang, W., Wang, X., Lu, H.: Visual tracking with fully convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition, Santiago, pp. 3119–3127 (2015)
9. Ma, C., Huang, J., Yang, X., Yang, M.: Hierarchical convolutional features for visual tracking. In: IEEE Conference on Computer Vision, Santiago, pp. 3074–3082 (2015)
10. Yeh, Y., Hsu, C.: Online selection of tracking features using AdaBoost. IEEE Trans. Circ. Syst. Video Technol. **19**(3), 442–446 (2009)
11. Godec, M., Roth, P.M., Bischof, H.: Hough-based tracking of non-rigid objects. In: IEEE International Conference on Computer Vision, Barcelona, vol. 117, no. 10, pp. 81–88 (2011)
12. Duffner, S., Garcia, C.: Fast pixelwise adaptive visual tracking of non-rigid objects. IEEE Trans. Image Process. **26**(5), 2368–2380 (2017)
13. Duffner, S., Garcia, C.: PixelTrack: a fast adaptive algorithm for tracking non-rigid objects. In: IEEE International Conference on Computer Vision, Sydney, pp. 2480–2487 (2013)
14. Chan, T.F., Sandberg, B.Y., Vese, L.A.: Active contours without edges for vector-valued images. J. Vis. Commun. Image Represent. **11**(2), 130–141 (2000)
15. Liu, L., Zhang, Q., Wu, M., Li, W., Shang, F.: Adaptive segmentation of magnetic resonance images with intensity inhomogeneity using level set method. Magn. Reson. Imaging **31**(4), 567–574 (2013)
16. Sun, X., Yao, H., Zhang, S., Li, D.: Non-rigid object contour tracking via a novel supervised level set model. IEEE Trans. Image Process. **24**(11), 3386–3399 (2015)
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556 [cs.CV] (2014)
18. Collins, R.T., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1631–1643 (2005)
19. Avidan, S.: Ensemble tracking. IEEE Trans. Pattern Anal. Mach. Intell. **29**(2), 261–271 (2007)
20. Tsai, D., Flagg, M., Rehg, J.M.: Motion coherent tracking with multi-label MRF optimization. Int. J. Comput. Vis. **100**(2), 1–11 (2010)
21. Liu, L., Fan, S., Ning, X., Liao, L.: An efficient level set model with self-similarity for texture segmentation. Neurocomputing **266**, 150–164 (2017)