# OBJECT LOCALIZATION FOR SUBSEQUENT UAV TRACKING

Diana B. Aglyamutdinova[1], Rail R. Mazgutov[1], Boris V. Vishnyakov[1]

[1] FGUP «State Research Institute of Aviation Systems», Russia, 125319, Moscow, Viktorenko street, 7 (agl.diana, mazgutov, vishnyakov)@gosniias.ru

**Commission II, WG II/5**

**KEY WORDS:** Tracking objects, initialization tracking algorithms, unmanned aerial vehicles (UAV)

## ABSTRACT:

The paper is devoted to the problem of semi-automatic initialization of the tracking algorithm, i.e. selecting an object of interest by unmanned aerial vehicles or drones. In this work, we propose an algorithm to refine the position and dimensions of the boundary box of the tracked object at the initial time (on the first frame), based on saliency detection algorithm, which simulates the map of human attention. We tested existing algorithms for object tracking by UAVs on the largest and most complex dataset – UAV 123. It is shown that the quality of tracking as a result of initialization by the proposed algorithm varies within acceptable limits for successful tracking of the object. The advantage of the proposed approach is that it applies the principles, used by the human visual system: the color, contrast, central focus.

## 1. INTRODUCTION

The recent progress in using UAVs for different human needs motivates software development researches in the fields of security and video surveillance. In security systems it is crucial to define the current position of the tracked object, so, applied to the UAV tracking problem, also to plan and form an optimal flight path of the unmanned aircraft in three-dimensional space. Therefore, a lot of computer vision teams work on the task of moving object detection and tracking from UAV in real-time.

A critical issue in tasks of visual tracking is the initialization or detection of an object of interest. The quality of tracking largely depends on it, as roughly defined position and size of the object of interest in the first frame entails rapid breakdown of the tracking process. Usually UAV operator marks an object of interest using his control pad, but due to wind, UAV speed and other disturbing factors, the result is mostly unsatisfactory for the future tracking. Bad initialization makes tracking a lot more difficult, because it leads to either the case when parts of the scene background occupy a significant part of the object's region (Vishnyakov et al., 2015), or the case when important parts of the object are discarded.

The task of semantic segmentation is also called scene parsing, it splits an image into semantically independent regions. It is also related to the object detection task. Such algorithms can be used to define the position and size of a traceable object. But they give redundant information in this situation, since we are only interested in the area, containing the traceable object. In addition, the algorithms of semantic segmentation are rather slow.

The first stage of the proposed approach is preliminary processing of the image (noise removal) by the Gaussian filter and converting the image into the CIE LAB color space. The next step is segmenting the image into homogeneous areas (superpixels) by the simple linear iterative clustering (SLIC) algorithm (Achanta et al., 2012).

## 2. MAIN BODY

### 2.1 Image pre-processing

The basic pre-processing task is noise reduction. Smoothing filters perform this task quite well. There are many linear and non-linear smoothing algorithms. Their usual application area is noise reducing, luminance stabilization, contrast and clarity enhancement. One of the popular smoothing methods is Gauss filtering. It has the successful application in many areas. Gaussian kernel coefficients are sampled from the 2D Gaussian function.

$$F(i,j) = \frac{1}{2\pi\delta^2}\exp\left(-\frac{i^2+j^2}{2\delta^2}\right) \qquad (1)$$

where σ is the standard deviation of the distribution,
$i, j$ – pixel coordinates.

We use 3x3 convolution kernel. Smoothed image is converted into the CIE LAB color space. The Lab color space describes mathematically all perceivable colors in the three dimensions: $L$ for lightness and $a$, $b$ for the color components green–red and blue–yellow respectively. The nonlinear relations for $L^*$, $a^*$, and $b^*$ are intended to simulate the nonlinear response of the human eye. Perceptual differences between any two colors can be approximated by taking the Euclidean distance between values in Lab color space. In our tests the algorithm showed better results using Lab than using RGB color space.

For the computational effectiveness homogeneous areas were used instead of discrete pixels.

### 2.2 Segmentation

Methods of segmentation as k-means method (Mirkes, 2011), watershed method (Beucher and Meyer, 1993), the method of graph cut (Boykov et al., 2001), simple linear iterative clustering (Simple Linear Iterative Clustering, SLIC) (Achanta et al., 2012) are able to break up the source image on different, but, in some sense, homogeneous areas named "superpixel" in a reasonable amount of time. SLIC method perform a local clustering of pixels in the 5-D space, defined by the $L, a, b$ values of the CIELAB color space and outputs better quality superpixels by a very low computational and memory cost.

SLIC segmentation algorithm:
1. Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]$ by sampling pixels at regular grid steps $s$.
2: Perturb cluster centers in an n × n neighborhood, to the lowest gradient position.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLII-2-9-2018

```
repeat
    for each cluster center $C_k$ do
        Assign the best matching pixels from a $2s \times 2s$ square neighborhood around the cluster center according to the distance measure (1).
    end for
    Compute new cluster center and residual error $E$ {$L_1$ distance between previous centers and recomputed centers}
until $E \leq$ threshold
Enforce connectivity.
```

All the pixels of image are allocated to clusters, referred to as ‘superpixels’ after segmentation algorithm. There is used non-oriented graph to store information about segments of image. Vertices of this graph are superpixels. Every vertex stores information about corresponding superpixel average color components, mean coordinates and it is on the boundary or not. Weight of the edges is a Euclidian distance between average colors of vertices.

Then, we need to calculate the object and the background measures.

## 2.3 Background measure

Background superpixels recognition is based on the idea that background regions have large perimeter on the boundary and object regions mostly have central location (Zhu et al., 2014).

Define geodesic distance $d_{geo}(p,q)$ as the shortest path between two vertices of superpixels graph. We calculate it using Johnson algorithm.

$$d_{geo}(p,q) = \min_{p=p1,\dots,pn=q} \sum_{i=1}^{N-1} d_{app}(p_i,p_{i+1}), \qquad (1)$$

where $p,q \in S$ – a set of superpixels, $|S| = N$
$B$ – set of boundary superpixels, $B \in S$,
$d_{app}(p,q) = dist(p,q)$ – a color contrast component, i.e. Euclidean distance between average color components of superpixels.

Let us define the boundary length:

$$\mathrm{Len}_{bnd}(p) = \sum_{q \in B} \exp\left(\frac{-d_{geo}(p,q)^2}{2\delta_{clr}^2}\right), p \in S \qquad (2)$$

where $\delta_{clr}$ – some color variation constant.

Bounding area of superpixel:

$$\mathrm{Area}(p) = \sum_{q \in S} \exp\left(\frac{-d_{geo}(p,q)^2}{2\delta_{clr}^2}\right), p \in S \qquad (3)$$

We can calculate background measure of superpixel $p$ as $w_{bg}(p)$, using (2) and (3):

$$w_{bg}(p) = \frac{\mathrm{Len}_{bnd}(p)}{\sqrt{\mathrm{Area}(p)}}, p \in S \qquad (4)$$

## 2.4 Object measure

In (Zhu et al., 2014) “Background weighted contrast” is used as an object measure.

$$w_{ctr}(p) = \sum_{p_i \in S} d_{app}(p,p_i)\, w_{spa}(p,p_i) w_{bg}(p_i), p \in S \qquad (5)$$

where the spatial component $w_{spa}(p,q)$ is:

$$w_{spa}(p,q) = \exp\left(\frac{-d_{spa}(p,q)^2}{2\delta_{spa}^2}\right)$$

where $d_{spa}(p,q)$ – Euclidean distance between centers of superpixels $p$ and $q$,
$\delta_{spa}$ – some spatial variation constant.

## 2.5 Saliency measure

The resulting saliency measure C(s) of a saliency map $s$, that we are trying to find, for each superpixel is calculated by optimizing the objective function value (Zhu et al., 2014), which combines background, foreground measures and a smoothing component:

$$\mathrm{C}(s) = \sum_{i=1}^{N} w_i^{bg} s_i^2 + \sum_{i=1}^{N} w_i^{fg} (s_i - 1)^2 +$$
$$+ \sum_{i=1}^{N} w_{ij} \left(s_i - s_j\right)^2 \to \min_s \qquad (6)$$

where $s$ – saliency map,
$i, j$ – indexes of superpixels $p_i, p_j$,
$w_i^{bg}$ – background measure,
$w_i^{fg} = w_{ctr}(p_i)$ – foreground (object) measure,
$w_{ij}$ – weight of the edge between adjacent superpixels in the graph:

$$w_{ij} = \exp\left(-\frac{d_{app}^2(p_i,p_j)}{2\delta_{nei}^2}\right) + \mu \cdot I(p_i,p_j),$$
$$I(p_i,p_j) = \begin{cases} 1, \text{if } p_i, p_j \text{ are adjacent} \\ 0, \text{else} \end{cases}.$$

To find the optimal values $\{s_i\}_{i=1}^{N}$ that minimize $C(s)$, we have to solve equation (6) using least squares method. Considering (8-11), the optimal value of saliency map $s$ can be found from (7):

$$A \cdot s = w_{ctr}(p_i), \qquad (7)$$

where $A = D - W + E^{bg} + E^{fg}$.

Component $W$ is the matrix of components $w_{ij}$ that defines the weights between superpixels $p_i$ and $p_j$.

$$W_{ij} = w_{ij} = w(p_i,p_j) \qquad (8)$$

Component D is the sum of the adjacent edges weights of superpixel:

$$D_{ij} = \begin{cases} \bar{w}(p_i), i = j \\ 0, i \neq j \end{cases} \qquad (9)$$

where $\bar{w}(p_i) = \sum_{p_j \in S} w(p_i,p_j)$.

Component $E^{bg}$ defines background measure.

$$E_{ij}^{bg}(i,j) = \begin{cases} \bar{w}(p_i), i = j \\ 0, i \neq j \end{cases} \qquad (10)$$

https://doi.org/10.5194/isprs-archives-XLII-2-9-2018

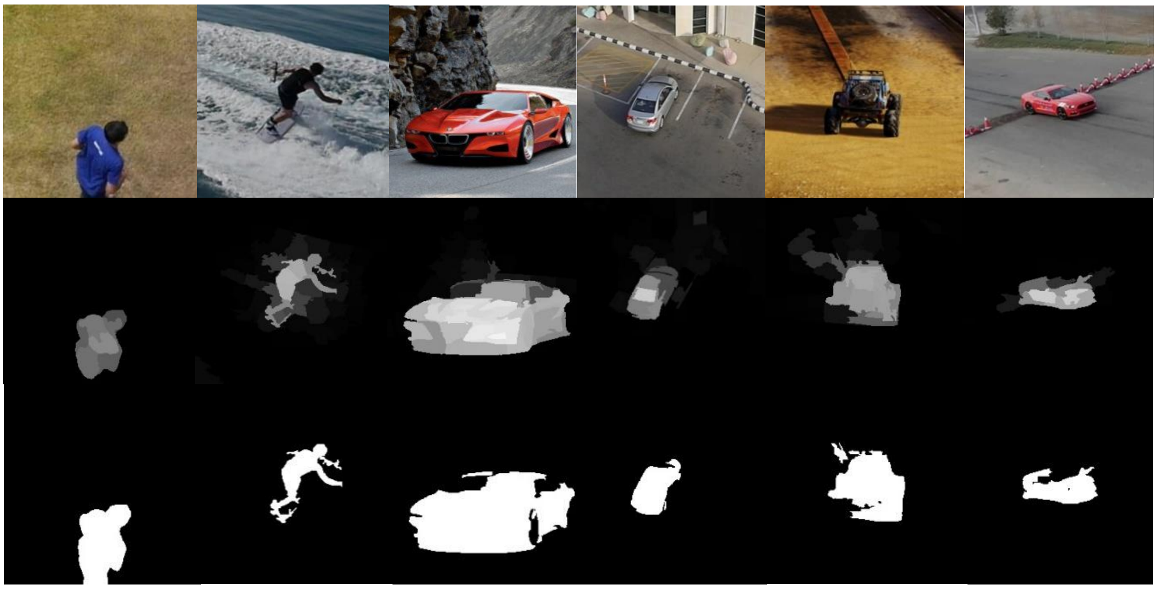


Figure 1. There are original images in the first row, saliency map in the second, binary saliency maps in the third

Component $E^{fg}$ defines foreground or object measure.

$$E_{ij}^{fg}(i,j) = \begin{cases} w_{ctr}(p_i), i = j \\ 0, i \neq j \end{cases} \qquad (11)$$

We used such values of hyperparameters in our experiments:
$\delta_{clr} = 7, \delta_{clr} = 0.4, \delta_{clr} = 5, \delta_{clr} = 10, \mu = 0.1.$

Each region in the generated saliency map is identified by values between 0 and 1, where the object of has values near 0 (marked white) as background has values near 1 (marked black).

## 2.6 Binarization

We convert the resulting saliency map into the binary image using binarization with an upper threshold:

Threshold can be found using Otsu method (Otsu, 1979).

$$s^*(i) = \begin{cases} 1, s(i) \geq t \\ 0, s(i) < t \end{cases}$$

## 2.7 Shadow removal

Then we perform shadow detection on image and remove shadow regions from object superpixels.
In this paper, shadow detection is based on the method (Blajovici, 2011), which uses the luminance statistics. Approach is based on the following considerations:
- the pixel belongs to the shadow when its brightness is less than 60% of the average brightness of the entire image.
- the pixel belongs to the shadow when its brightness is less than 70% of the average brightness in a superpixel.

## 2.8 Binary image processing

Binarization results may lead to small objects that lay outside the target object or target object can be divided into parts. Therefore, we need to delete some needless separate elements and bring parts of the foreground together.

For small target objects (width and height of 5-20 pixels) we use erosion operation to small fragments with a structuring element in the form of a circle (having two-pixel radius). As a result, all found objects will be reduced in size. To restore the shape of objects, the dilating operation with the same structuring element is then used. Next, to connect the small parts into one, a dilating operation with a circle (having three-pixel radius). All constants, mentioned above, may vary for a target object of different size.

## 2.7 Experiments

By the next step we apply described approach to initialize a number of fast and effective methods of tracking object: "DCF_CA" (Mueller et al., 2017), "MOSSE_CA" (Mueller et al., 2017), "SAMF" (Kristan et al., 2014), "DCF" (Henriques et al., 2012), "DSST" (Danelljan et al., 2014), "MOSSE" (Bolme, 2010), "SRDCF" (Danelljan et al., 2015). In this way we test the quality of the algorithm on the largest and complex database of video clips taken with unmanned aerial vehicle – UAV 123.

The scores for these trackers are based on two metrics, precision and success rate (Table 1). Precision is measured as the distance between the centers of a tracker bounding box and corresponding ground truth bounding box. The precision plot shows the percentage of tracker bounding boxes within a given threshold distance in pixels of the ground truth. To rank the trackers, we use a threshold of 20 pixels (Bolme et al., 2010). The success is measured as intersection over union of pixels in tracker bounding box and corresponding ground truth bounding box.

$$Success = \frac{b_{gt} \cup b_{tr}}{b_{gt} \cap b_{tr}}$$

where $b_{gt}$ – ground truth bounding box,
$b_{tr}$ – tracker bounding box.

For initialization of the tracker we use:
1) triple-sized ground truth region with our saliency algorithm, predicting the initialization region of an object;
2) ground truth region.

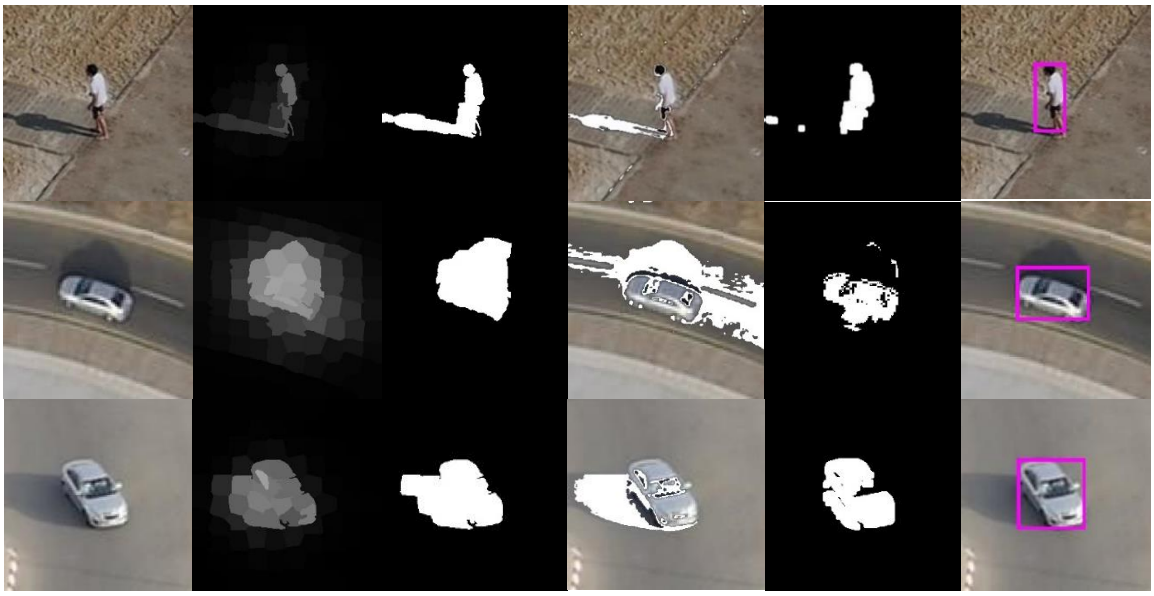


Figure 2. There are original images in the first column, saliency map in the second, binary saliency maps in the third, object shadows, detected on the image in the fourth column, binary image without shadow in the fifths and bounding boxes in sixths.

| Tracking algorithm | Average FPS | Success difference | Precision difference |
|---|---|---|---|
| DCF_CA | 182.2102 | 0.176 | 0.176 |
| MOSSE_CA | 271.7864 | 0.064 | 0.065 |
| SAMF | 8.5000 | 0.216 | 0.182 |
| DCF | 238.9312 | 0.160 | 0.154 |
| DSST | 130.7184 | 0.155 | 0.165 |
| MOSSE | 253.6889 | 0.136 | 0.124 |
| SRDCF | 8.9679 | 0.234 | 0.175 |

Table. 1. Trackers average FPS, success rate difference for semi-automatic and ground truth initialization with IoU > 0.5 and precision difference for semi-automatic and ground truth initialization with 20-pixel precision threshold.

The success plot (Table 2) shows the percentage of tracker bounding boxes whose overlap score $S$ is larger, than a given threshold.

## 3. CONCLUSIONS

According to the results of the experimental testing we can conclude that the best tracking quality in the initialization of the proposed algorithm is achieved by tracking algorithms "SRDCF" and "MOSSE_CA". It is easy to notice that the tracking algorithm "MOSSE_CA" outperforms other algorithms according to the experiments. Thus, the most appropriate algorithm for tracking objects from UAVs combined with the proposed algorithm of the initialization is "MOSSE_CA", because it was least sensitive to the accuracy of initialization and it is the most fast-acting in comparison with its competitors.

The proposed algorithm does not require special hardware and can work in real-time. It is implemented in C++. The average time required before the object is specified, occupying 40% of the image size $256 \times 256$ pixels, is equal to 60 milliseconds on the Intel® Core ™ i5-3470 CPU @ 3.20GHz.

## 4. ANKNOLEGMENTS

This work was supported by Russian Science Foundation (RSF) under Grant 16-11-00082; by Russian Foundation for Basic Research (RFBR) under Grant 16-08-01260 A.

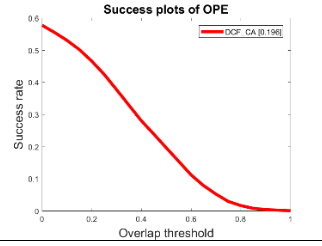
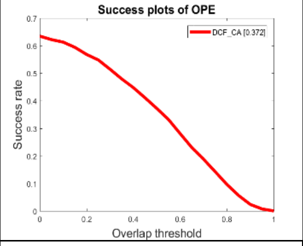
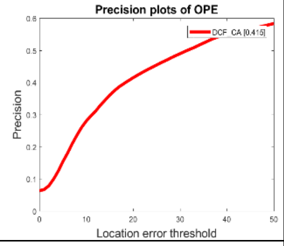
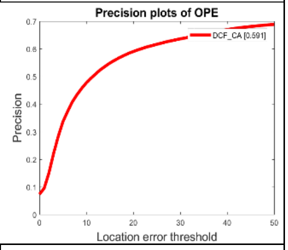
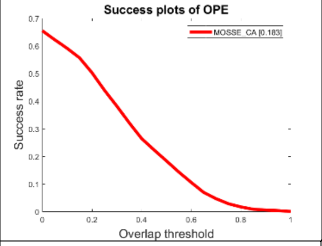
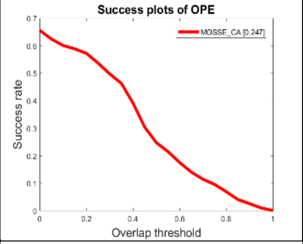
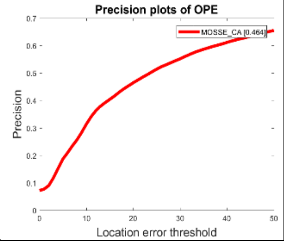
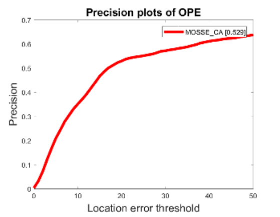
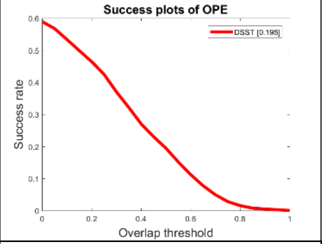
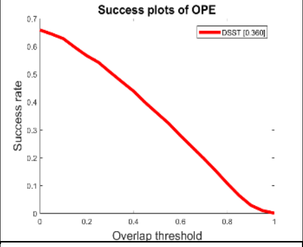
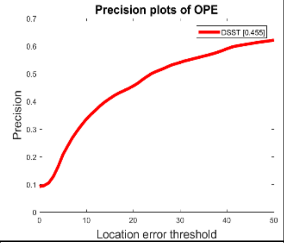
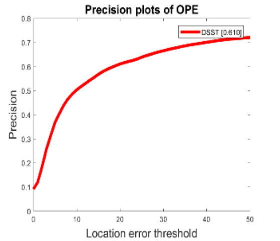
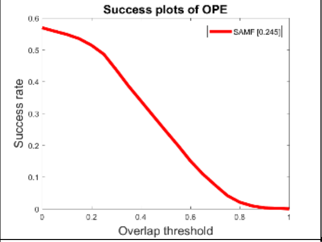
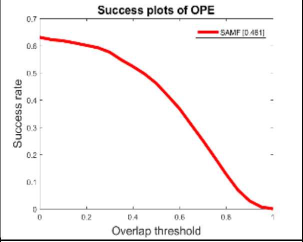
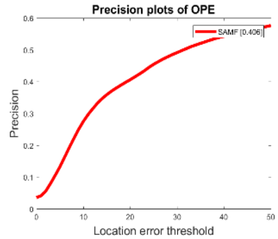
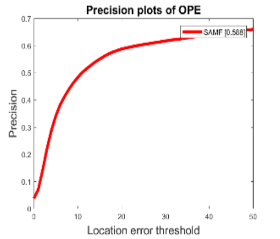
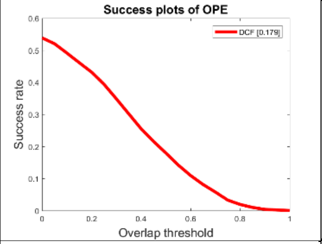
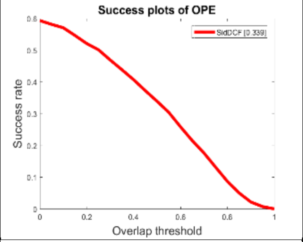
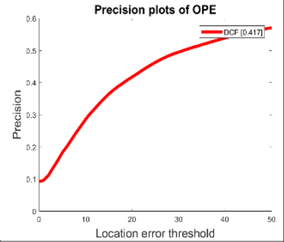
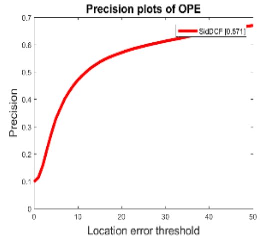
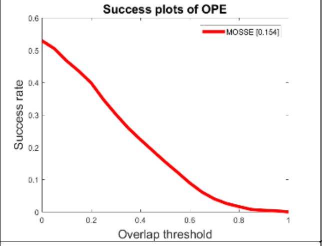
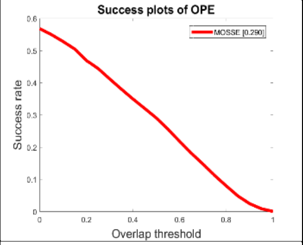
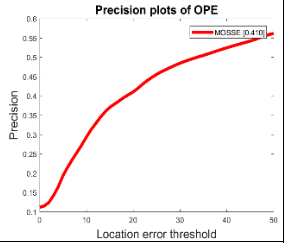
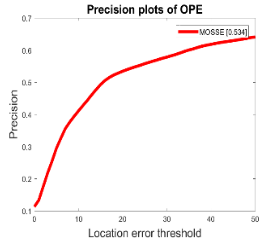
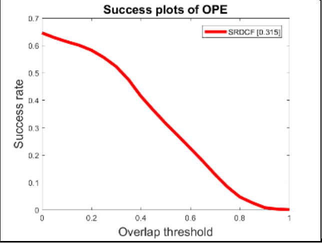
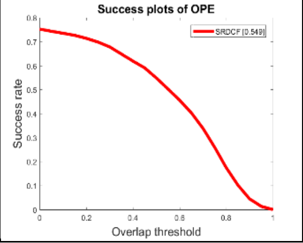
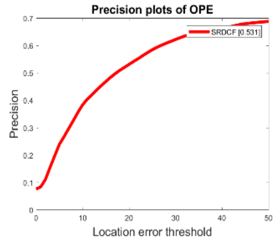
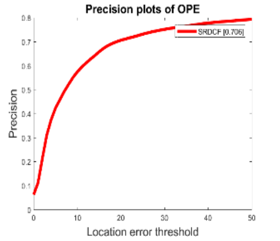
| Tracking algorithm | Success plot with semi-automatic initialization | Success plot with ground truth initialization | Precision plot with semi-authomatic initialization | Precision plot with ground truth initialization |
|---|---|---|---|---|
| DCF_CA |  |  |  |  |
| MOSSE_CA |  |  |  |  |
| DSST |  |  |  |  |
| SAMF |  |  |  |  |
| DCF |  |  |  |  |
| MOSSE |  |  |  |  |
| SRDCF |  |  |  |  |

Table 2. Success and precision graphs of different tracking algorithms with semi-automatic and ground truth initialization.

## 5. REFERENCES

Achanta R., Shaji A., Smith K., Lucchi A., Fua P. and Susstrunk S., 2012. "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods" In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol 34 No 11. pp 2274-2281, November 2012.

Beucher Serge, Meyer Fernand, 1993. The morphological approach to segmentation: the watershed transformation. In: *Mathematical Morphology in Image Processing (Ed. E. R. Dougherty)*, pp. 433-481.

Blajovici C., Kiss P. J., Bonus Z., Varga L., 2011. Shadow detection and removal from a single image. In: *SSIP*, Hungary, pp.1-6.

Bolme, D.S., Beveridge, J.R., Draper, B.A., Lui, Y.M., 2010. Visual object tracking using adaptive correlation filters. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on.* pp. 2544-2550.

Boykov, Y., Veksler, O., and Zabih, 2001. R. Fast approximate energy minimization via graph cuts In: *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(11): pp. 1222-1239.

Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M, 2014. Accurate scale estimation for robust visual tracking. In: *Proceedings of th e British Machine Vision Conference. BMVA Press.*

Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M., 2015. Learning spatially regularized correlation filters for visual tracking. In: *The IEEE International Conference on Computer Vision (ICCV).*

Grady Leo, 2006. Random Walks for Image Segmentation In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1768-1783.

Henriques, J.F., Caseiro, R., Martins, P., Batista, J., 2015. High-speed tracking with kernelized correlation filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (3), pp. 583-596.

Kristan, M., Pflugfelder, R., Leonardis, A., Matas, J., Cehovin, L., Nebehay, G., Vojır, T., Fernandez, G., Lukezic, A., Dimitriev, A., et al., 2014. The visual object tracking vot 2014 challenge results. In: *Computer Vision-ECCV 2014 Workshops*. pp. 191-217. Springer.

Kulchin Yu.N., Notkin B.S., Sedov V.A., 2009. Neuro-iterative algorithm of tomographic reconstruction of the distributed physical fields in the fibre-optic measuring systems. *Computer Optics.* T. 33, № 4. pp. 446-455. (in Russian).

Mirkes E.M., 2011. K-means and K-medoids applet. *University of Leicester.*

Mueller Matthias, Smith Neil, Ghanem Bernard, 2017. Context-Aware Correlation Filter Tracking In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1387-1395.

Otsu N., 1979. A threshold selection method from gray-level histograms. In: *IEEE Trans. Sys., Man., Cyber.* 9: 62-66.

Vishnyakov Boris, Sidyakin Sergey, Vizilter Yury, 2015. Diffusion background model for moving objects detection Moscow, Russia. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume XL-5/W6, pp. 65-71.

Wu Y., Lim J. and Yang M.-H., 2013. Online Object Tracking: A Benchmark, In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2411–2418.

Zhu W., Liang S., Wei Y., Sun J., 2014. Saliency Optimization from Robust Background Detection In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2814-2821.