Maritime Vessels Real-time Tracking-by-detection in UAV Videos

Pavel A. Kazantsev*, Victor A. Sadakov and Maxim V. Chupakov

Federal State Higher Military Educational Establishment, Nakhimov Black Sea Higher Naval School, Sevastopol, Russia; pak@pawlin.ru, vialsa@yandex.ru, maxim.vor2012@yandex.ru

Abstract

Background/Objectives: This paper presents a fast real-time multi-object tracking algorithm adopted for maritime vessels tracking in UAV videos. **Methods/Statistical Analysis:** This research applies extended multi-object tracking-by-detection framework in highly dynamic UAV-captured maritime environment. Propagation is performed using particle filter whose particle weights are updated using modified observation model that incorporates a term based upon trackers affinity. The latter is also used for trackers grouping, that handles detector inaccuracies, and preventing identity switches by means of special propagation mode that is enabled when targets approach each other and start to overlap. **Findings:** Our research has shown that state-of-art multi-tracking algorithm is applicable to maritime vessels real-time tracking in UAV videos, provided the use of weak and fast online classifiers, which weakness is compensated by algorithm features based on trackers affinity matrix. **Improvements/Applications:** This paper presents an approach to maritime vessels tracking that allows to use fast, but less accurate, simple detectors and classifiers, enabling real-time processing on board of small-sized UAVs, while keeping decent precision and accuracy.

Keywords: Maritime Vessels Tracking, Multi-object Tracking, Particle Filter, Online Learning, Trackers Affinity, Trackingby-detection, UAV Surveillance

1. Introduction

Unmanned Aerial Vehicle (UAV) video surveillance is an area of active research in the last decade¹⁻³. This holds true for maritime vessels tracking using various non-aerial video surveillance systems, as covered in a survey⁴.Few research teams addressed a problem of maritime object detection in full-motion UAV video^{5.6}. However, maritime vessels tracking in UAV video is underrepresented in scientific literature. In⁶ tracking block is included into algorithm block diagram, but is not covered in details, except that authors mention the use of Kalman filter for prediction. The lack of articles dedicated to this topic can be explained by specifics of this area of application⁶. These specifics that boil down to non-stationary and dynamic behaviors of ocean surface, prevent the effective use of background subtraction and motion detection techniques. Abrupt scene and perspective changes, caused

by non-linearity of UAV flight path, add to the problem above plus lowers precision of detection-only tracking.

Multi-object tracking algorithms, however, are very well presented in literature, since it is an object of active research for two decades already. These works mostly address moving objects tracking in urban environments. State-of-art multi-object tracking algorithms also handle issues inflicted by moving platform, crowded scenes, illumination variability, and use tracking-by-detection approach supported by object contents information in order to propagate trackers 7-10. There two main strategies for tracking: online tracking^{7.8} and offline tracking^{9.10}. In online tracking approach data is associated using only preceding information about trackers and detections, while in offline approach a global optimization is performed by taking into account future and past entries. Offline algorithms show the best accuracy results when tested on most challenging benchmark¹¹. However, online algorithms still have their niche of applications, like UAV surveillance, where prompt results are required.

Among search algorithms, that estimate the most probable location for a tracker at any given time, particle filters¹³ and dense sampling¹⁴ show the best results for single-object tracking¹². Dense sampling, however, is a computationally intensive procedure^{15,16}. Particle filter is relatively robust to noises and much less time-consuming than dense sampling^{17,18}. Since in UAV on-board applications computational efficiency is a key factor, we use particle filter in our tracking algorithm.

In¹⁹ we introduced a tracking-by-detection algorithm that makes use of velocity, size and location of trackers in order to measure affinity between them. This additional information is fed to several blocks in our pipeline in order to balance accuracy losses caused by weaker (but faster) versions of detectors and model classifiers.

In this work, as our contribution, we adopt state-of-art tracking-by-detection framework, based upon detections, representation models and trackers affinity matrix, to maritime vessels tracking in UAV videos. To the best of our knowledge we are first to do so.

2. Concept Headings

2.1 Algorithm Overview

In our previous research¹⁹ we used tracking-by-detection framework proposed in^z and extended it by incorporation of trackers' relationships between each other, based on their locations, sizes and velocities. These characteristics play a key role in a decision-making in several algorithm steps. Firstly, they are used to group trackers. Grouping procedure takes these characteristics into account in order to make clusters of trackers, followed by conditional merging of these trackers into one entity. This mechanism alleviates a problem that arises in these cases, where several trackers are initiated for one actual object. It happens when detections are triggered on an object inaccurately, that leads to the scenario where multiple trackers latch to the same target. But in this case, trackers' locations and velocities will become nearly equal after some time steps. For UAV applications, performance is a critical issue, especially in online tracking, and trackers grouping speeds up an algorithm by removing redundant trackers that would otherwise take an extra computation time to propagate. Relationship between two trackers is calculated as an affinity score and put into respective cell of the *trackers affinity matrix*.

Affinity scores are used in particle filter's weights update rule in a form of *anti-weight* – an extra term that always comes with a negative sign. Anti-weights almost nullify weights of those particles, that get inside of the bounding boxes of non-affine trackers, i.e. trackers which physical characteristics differ greatly. This addition to observation model makes identity switches less probable, especially for objects with similar shape, size and color, like it is shown in Figure 1. This feature plays even a bigger role when operating with weak (but fast) representation models.



Figure 1. Vessels with similar shape, size and color that would yield almost identical representation models.

If a value, obtained by averaging all of the anti-weights for particular tracker, exceeds a certain threshold, a *bad tracking case mode* is enabled for this tracker. This mode corresponds to scenarios where targets overlap (completely or partially). When bad tracking case mode is enabled, approximated location becomes the only source of information that guides a tracker, since detections and classifiers can't be relied upon during such an event. We introduced this mode in¹⁹ and tested it mostly on pedestrians in cluttered scenes, where it proved to be effective. In UAV videos, top views are encountered the most, and with such views targets can't overlap, so this mode may seem useless. Nevertheless, side views are also in place, especially for remote targets, like it is shown in Figure 2, so, we keep this mode for this application as well.



Figure 2. Overlap situation occurred during a side view of remote marine vessels.

Though effects on water surface pose certain problems to object detection, we still can utilize its image properties in order to guide trackers. To do so, we build an approximate water mask using simple clustering algorithm (see Section 2.5), and use this mask in order nullify likelihoods of those particles that propagated to water. This mask need not to be precise and account for glints, foam trails and other water surface effects, since its role is auxiliary, unlike masks used for blob detection.

2.2 Particle Filter

We keep particle filter as our search algorithm, as we did in¹⁹. That being said, target state $X = \{x, y, v_x, v_y\}$ is given by coordinates (x, y) and velocity components (v_x, v_y) . Propagation model is given by:

$$(x, y)_{t} = (x, y)_{t-1} + (v_{x}, v_{y})_{t-1} \cdot \Delta t + \delta(x, y)$$

$$(v_{x}, v_{y})_{t} = (v_{x}, v_{y})_{t-1} \cdot \Delta t + \delta(v_{x}, v_{y})$$
(1)

Coordinate and velocity noises, $\delta(x, y)$ and $\delta(v_x, v_y)$, respectively, are obtained from zero-mean normal distributions. Components of position variance $\sigma(x, y)$, σ_x and σ_y , are derived from tracker's width and height, respectively. Velocity variance $\delta(v_x, v_y)$ is given by:

$$\delta(v_x, v_y) = \chi(v_x, v_y) + \tau(v_x, v_y)$$
(2)

where $\chi(v_x, v_y)$ is a noise velocity that is parallel to tracker's current velocity vector (centripetal noise) and $\tau(v_x, v_y)$ - noise velocity orthogonal to tracker's current velocity vector (tangential noise). Their values depend on what abrupt movement patterns are to be handled. Large acceleration values (both, positive and negative) are better handled with bigger centripetal variance, while abrupt change of direction is meant to be handled by bigger tangential variance. In UAV videos abrupt changes of direction are more probable, that is why tangential variance is of a bigger value for this particular application.

Tracker's life span is determined by initialization, termination and renewal rules that are the same as we used in¹⁹.

2.3 Vessel Detection

For vessel detection we use an algorithm, combined from blocks we previously used in other works for detection of various objects. We start with regions of interest extraction by applying MSER algorithm²⁰ to down-scaled image, just like we did in²¹ for pedestrian detection in RGB-D data. MSER algorithm extracts blobs followed by approximation by ellipses. Then, ellipses are approximated by bounding rectangles that undergo simple grouping and filtering by size. Size filtering is used only to remove obviously large rectangles which dimensions are almost equal to image size. Region of interest extraction example is shown in Figure 3.



(a)



Figure 3. Region of interest extraction: a) maximally stable extremal regions approximated by ellipses b) final regions of interest obtained after grouping and filtering of rectangles that approximate ellipses.

Regions of interest are fed into algorithm based on deep autoencoders we proposed in²² for head-shoulder detection, but retrained on vessel imagery from CIFAR-10 database²³ and enhanced by regularization procedures that provide invariance to orientation and shifts, proposed by us in²⁴ and applied in²¹.Each region is processed separately and can be a host to several detections or lack them completely. For regions illustrated on Figure 3, final detections and trackers initialized by them, are shown on Figure 4.

2.4 Data Association

In order to match detection d with a tracker tr we use *greedy data association* algorithm^Z. General formulae for matching score is the same as used in^Z :

$$S(tr,d) = g(tr,d) \cdot \left(c_{tr}(d) + \alpha \cdot \sum_{p \in tr}^{N} p_{N}(d-p) \right), \qquad (3)$$

where terms $c_{tr}(d)$ and $\sum_{p=r}^{N} p_N(d-p)$ are representation model score and the normal distribution $N(pos_d - pos_p; 0; \sigma^2)$ calculated as a distance between the center of detection *d* and a particle *p*, respec-

tively. A gating function g(tr, d) is given by:

$$g(tr,d) = \begin{cases} p_{N}\left(\frac{size_{tr} - size_{d}}{size_{tr}}\right) \cdot p_{N}\left(\left|d - tr\right|\right) & (4) \\ 0 & \end{cases}$$

if
$$B(tr) = false$$
 otherwise

where B(tr) is an Boolean function that returns *true* if a bad tracking mode is enabled (see Sec. 2.7) for particular tracker. Data association or classifier update are not allowed in this mode.

Online combined classifier training technique, proposed by us in¹⁹, appeared to be quite effective for UAV applications, where performance is of a great value, if we want to process all tracking pipeline on-board. The main difference from analogous techniques, proposed in⁸ and used in⁷, consists in that we don't retrain classifier on each frame and do it only when its quality measure drops below a certain threshold. It makes classifier weaker, but much faster. Such a drop in quality is compensated by observation model's anti-weights, as described in Section 2.5.



Figure 4. Applying autoencoder-based detector: a) detector output b) initialized trackers.

In order to merge different classifiers (color and texture) we transform distances between calculated and stored models into probabilities and bring it to the range of [0; 1] by formula:

$$P = \exp\left(-\frac{(d-d_m)^2}{2\cdot\sigma^2}\right)$$
(5)

where, P - score (probability) estimate for particular model in the range of [0;1]; d - distance between detection model and tracked target model in corresponding space; d_m - minimal possible distance for this model, σ - sigma for this model.

2.5 Observation Model

Observation model, in general, is kept the same as in^{19} , but got some rework in order to tailor it to maritime scenario:

$$w_{r,p} = \begin{cases} \beta(d_r) \cdot p_N(p - d_r) + \eta \cdot c_r(p) - \mu \cdot w_a(p) & B(r) = false \\ p_N(p - d_{est}) & B(r) = true \\ 0 & M(p) = 0 \end{cases}$$
(6)

where, η and μ are fixed parameters and remain unchanged during entire video-sequence. B(*tr*) is the same indicator function as in (4). Terms of (6) are described below. First two cases in (6) are the same as we used in observation model in¹⁹, and the third one is a new, tailored specifically for maritime application.

Water surface mask M is obtained by clustering pixels of downscaled image into two clusters with Expectation Maximization (EM) algorithm²⁵. Pixels of largest cluster are considered to be water surface and correspond to zero values in mask M (see Figure 5). Hence, according to (6), weights of particles that propagate to locations, where water mask is zero, are turned to zero. The reasoning behind this technique consists in that we restrict particle propagation to unoccupied area, thus increasing sampling density in the vicinity of objects. When a vessel is firmly detected, water surface mask makes tracking a bit more accurate. But when detection is amiss or when abrupt point of view occurs (due to UAV surveillance nature - maritime vessels themselves are very inertial objects and don't produce abrupt movement patters), water surface mask can be a problem solver. Surely, this mask doesn't correspond to water surface precisely, as can be seen from Figure 5, and even can be worse than that due to glitters and other illumination effects. But since it

is used as auxiliary information for probabilistic approach (and particle filter search algorithm is the one), reducing potential search area even by mere percents can only help, provided computing time it takes to build that mask is worth it (that is why we perform EM on downscaled image).



(a)



Figure 5. Extracting water surface mask: a) original image b) water surface mask.

Detection term $\beta(d_{tr}) \cdot p_N(p-d_{tr})$ is computed as normal distribution P_N from the distance between the center of detection d_{tr} and location of the particle *P*. d_{tr} is defined by:

$$d_{tr} = \begin{cases} d_a \\ d_{est} \end{cases}$$
(7)

if association occurred

otherwise

where, d_a - associated detection, and d_{est} - approximated location of the tracker with no associated detection. Location approximation is done by transitioning its previous location using average velocity vector:

$$\vec{v} = \frac{\vec{P}_{t-1} - \vec{P}_{t-t_0}}{\Delta s} \tag{8}$$

where, t - is a current time, t_0 - time gap for averaging (approximately one real second), Δs - a distance a tracker has traveled during $[t-1,t-t_0]$ interval.

Classifier term $\eta \cdot c_r(p)$ is a representation models similarity score, described in Sec. 2.4. This score is computed in a location of particle p using current rectangle of a tracker.

Anti-weight term $\mu \cdot w_a(p)$ is a negative term and, thus, reduces resulting weight $w_{tr,p}$. Anti-weight is computed using formulas:

$$w_a(p_{tr}) = \max_{tr' \in T, tr' \neq tr} \left\{ w_a(p_{tr}, tr') \right\}$$
(9)

$$w_{a}\left(p_{\nu}, tr'\right) = \begin{cases} \left(1 - R\left(tr, tr'\right)\right) \cdot p_{N}\left(p_{\nu} - tr'\right) & R\left(tr, tr'\right) < \rho \\ 0 & R\left(tr, tr'\right) \geq \rho, \quad p_{\nu} \in box_{\nu} \end{cases}$$

$$R\left(tr, tr'\right) = \frac{g\left(tr, tr'\right) \cdot \left(\gamma \cdot p_{N}\left(tr - tr'\right) + \upsilon \cdot p_{vel}\left(tr, tr'\right) + \vartheta \cdot p_{area}\left(tr, tr'\right)\right)}{R_{max}}$$

$$(11)$$

R- is a $M \times M$ trackers affinity matrix and R(tr, tr') is an affinity score of the pair of trackers, which explanation is given in our preceding work¹⁹. Gating function g(tr, tr') is given by:

$$g(tr,tr') = g_{sp}(tr,tr') \cdot g_a(tr,tr')$$
(12)

 $g_{sp}(tr,tr')$ - speed gating function that is given by:

$$g_{sp}(tr,tr') = \begin{cases} 0 & |v_{tr}| < \tau_v \text{ and } |v_{tr'}| > \tau_v \\ 0 & |v_{tr'}| < \tau_v \text{ and } |v_{tr}| > \tau_v \\ 1 & otherwise \end{cases}$$
(13)

where, v_{r}^{avg} and v_{r}^{avg} are average velocities given by (8); τ_{v} - is a speed threshold under which a target is considered stationary.

 $g_a(tr, tr')$ - angle gating function that is given by:

$$g_{a}(tr,tr') = \begin{cases} 0 & \arccos\left(\frac{v_{tr} \cdot v_{tr'}}{|v_{tr}| \cdot |v_{tr'}|}\right) > \varphi_{v} \\ 1 & otherwise \end{cases}$$
(14)

 $p_{\rm N}(tr-tr')$ from (11) is a normal distribution taken from distance between centers if trackers tr and tr'. $p_{vel}(tr,tr')$ is by (15) and $p_{area}(tr,tr')$ is given by (16).

$$p_{vol}(tr,tr') = \begin{cases} 1 & |v_{\nu}| < \tau_{v} \text{ and } |v_{\nu'}| < \tau_{v} \\ p_{N}\left(|v_{\nu}| - |v_{\nu'}|\right) \cdot p_{N}\left(\arccos\left(\frac{v_{\nu'}v_{\nu'}}{|v_{\nu'}| + |v_{\nu'}|}\right)\right) & otherwise \end{cases}$$
(15)

$$p_{area}(tr,tr') = p_{\rm N}\left(\frac{\min(s_{tr},s_{tr'}) - S_{tr \cap tr'}}{\min(s_{tr},s_{tr'})}\right)$$
(16)

3. Results

Due to the underrepresentation of algorithms for maritime vessel tracking in UAV video in scientific literature, currently there are no benchmark datasets available. In order to assess performance of our algorithm we used various UAV videos with maritime vessels, found on Youtube and captured with Phantom 3 and 4 drones²⁷. These videos feature the following challenges: occlusions, non-linear movement patters (usual for any UAV application), changes of point of view, water surface glitters and unstable illumination effects, foam trails. In these videos UAV operator usually focuses on some main ship (see a ferry on a forefront on Figures 2-5), which is an objective of the filming, with some secondary targets appear in the vicinity occasionally, but for a prolonged periods of time that allow actual tracking. More examples of such videos can be found in²⁸. To evaluate precision and accuracy we use CLEAR MOT metrics²⁹ and do it in a way just like we did with pedestrian targets in¹⁹. In order to obtain ground truth we've marked up videos manually. Our runtime performance results were obtained on workstation with Intel Core2Duo 3GHz with 4GB of memory and on Raspberry Pi board with Quad-Core ARM 1,2 GHz processor²⁸, that can be installed on light-weight UAVs.

Table 1 shows evaluation results averaged for 23 videos taken from²⁸. We separated results for main and secondary ships tracking, since they differ too much. FP rate, though, bears a common portion, caused by false detects. Table 2 shows main parameters of our algorithm that affect accuracy and computational performance the most.

4. Discussion

In this section we discuss the nature of inaccuracies observed during experiments and their effect on

Dataset	МОТР	МОТА	FN	FP	ID sw.	FPS PC	FPS Raspberry Pi 3
Main ship (filming objective)	83.2%	92.1%	2.3%	5.6%	5	23	11
Secondary targets	75.2%	70.9%	15.2%	13.9%	26	-	-

Table 1. CLEAR MOT evaluation results

 Table 2. Main algorithm parameters that affect accuracy and computational performance

Parameter	Value	Comment	
Number of particles	100		
α	5	With this target is being tracked by detection term in (3) if detection actually occurs. Representation model steps into play when detection is amiss.	
β:η:μ	10:1:20	With that, in accordance with observation model (6), propagation is done mostly by location likelihood. Anti-weight coefficient μ makes sure that weights of those particles, that propagate into non-affine trackers bounding box, are nullified.	
Downscale coefficient for MSER regions of interest extraction	4	MSER algorithm is $O(n \cdot \log(\log(n)))$ algorithm. Down-scaling the image with coefficient s, reduces number of pixels by s ² , hence, with s << n, virtually making it s ² faster.	
Downscale coefficient for EM water surface mask	4	EM is $O(n \cdot k)$ for each iteration, where k is a number of clusters. With downscaling coefficient of s, we speed up EM by s ² . Again, since water surface mask is used as auxiliary source of information that makes particles propagation more stable.	
γ:υ:θ	1:2:1	This makes velocity more influential in affinity score. With that, when a ship falls into two or more detections, these detections will be following the same rigid body anyway (which means similar velocities), while staying apart from each other for several frames. With this coefficient ratio they'll have a higher chance to become affine and merge.	

numerical results, presented in Table 1. Main ship tracking is rather firm in comparison with secondary targets for obvious reasons. First of all, main ships are firmly detected, because they always appear on forefront and their appearances bear distinctive features, extracted during deep learning with autoencoders using CIFAR-10. Nevertheless, sometimes, as mentioned above, they fall into several (mainly two) detections. It mostly affects MOTP metric (non-precise localization), though, since due to tracking grouping, these detections merge after few frames, thus, FP metric is not affected much. Low (but not-perfect) FN rate for main ships caused by relatively low false reject rate of our detector. Couple of identity switches are caused by losing a sight of main ship and failures of re-entry association.

Secondary ships that mostly appear in the background or in docks/ports are both harder to detect and track. In many cases secondary ships are being led by representation model only after they got spotted by detector once at some lucky moment. It inputs the most into FN rate. False positives are caused by two reasons. First reason concludes in false detects, that mostly occur on landscapes, that appear in UAV's sight, when main ship leaves or arrives to the port (this fraction of FP is shared between main and secondary ships statistics). The second reason, exclusive for secondary ships only, is caused by floating trackers that lose their targets due to absence of detection and instability of representation models - both are common for small targets. FP and FN metric would be even worse if not for anti-weight term, that prevents trackers from jumping on targets that are already occupied by non-affine trackers. Largest input to identity switches for secondary targets is caused by premature deactivation of trackers due to lack of detections and stable representation models that would otherwise rejuvenate trackers. Also, few identity switches for secondary targets are caused by occlusions that are not always properly handled by bad tracking case mode technique.

5. Conclusion

We have introduced an algorithm for fast multiple maritime vessels tracking in UAV videos. To the best of our knowledge it is a first attempt to apply state-of-art tracking-by-detection framework in this particular highly dynamic environment. This work is based on our previous research aimed at urban (pedestrians and cars) objects tracking, but features several enhancements peculiar to maritime UAV applications. This attempt proved to be rather successful both accuracy and computation performance-wise, since our previous work was aimed at developing a tracking algorithm that would yield competitive performance even with weak (hence fast) detectors and representation models. Tracking-by-detection framework, with particle filter as a search algorithm, allows easy switching between various different applications. This flexibility is largely due to probabilistic models for data association and observation. By redefinition of existing and introduction of new specific likelihood terms in respective formulas one can manipulate trackers behavior, depending on application. Application-specific process noise in particle filter propagation rule is also adjustable for expected movement patterns. In order to provide fast tracking, suitable for real-time implementation on UAV, we utilize weak, but fast, online classifiers, whose weakness is compensated by features based on trackers affinity score. Overall, our algorithm can be enhanced by adding a backup blob detector, separating landscape and skies (by horizon line detection, for instance) from water surface.

6. Acknowledgement

The authors thank Ministry of Education and Science of the Russian Federation for supporting this research. Funding Information. The research was financed by the Ministry of Education and Science of the Russian Federation under the Grant Contract dated 27 October, 2015, (grant number 14.607.21.0127, contract unique identifier RFMEFI60715X0127), the grant was assigned for the research into "Development and implementation of autonomous light-weight aircraft UAV shipdeck landing using machine vision system".

7. References

1. Ali S, Shad M. COCOA - Tracking in aerial imagery airborne intelligence surveillance reconnaissance ISR systems and applications III. SPIE; 2006:D2090.

- Yu W, Yu X, Zhang P, Zhou J. A new framework of moving target detection and tracking for UAV video application. ISPRS08 Congress, Beijing, China, 2008 Jul 3–11; 2008.
- Mattyus G, Benedek C, Sziranyi T. Multi target tracking on aerial video. ISPRS Instanbul Workshop 2010 on Modeling of optical airborne Sensors, WG I/4, IAPRS Vol. XXXVIII, part 1/W4, 2010 Oct 11–13; 2010.
- Moreira RD, Ebecken NFF, Alves AS, Livernet F, Campillo-Navetti A. A survey on video detection and tracking of maritime vessels. International Journal of Signal Processing. 2016; 1:47–60.
- Marques JS, Bernardino A, Cruz G, Bento M. An algorithm for the detection of vessels in aerial images. 2014 11th IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS), Seoul; 2014. p. 295–300. DOI: 10.1109/AVSS.2014.6918684.
- Parameswaran S, Lane C, Bagnall B, Buck H. Marine object detection in UAV full-motion video. SPIE 9076, Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications XI, 907608; 2014. DOI: 10.1117/12.2049487.
- Breitenstein MD, Reichlin F, Leibe B, Koller-Meier E, Van Gool L. Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2011; 33(9):1820–33. DOI: 10.1109/TPAMI.2010.232.
- Grabner H, Leistner C, Bischof H. Semi-supervised on-line boosting for robust tracking. Computer Vision – ECCV. Volume 5302 of the series Lecture Notes in Computer Science; 2008. p. 234–47.
- 9. Choi W. Near-online multi-target tracking with aggregated local flow descriptor. ICCV; 2015.
- Wang B, Wang G, Chan KL, Wang L. Tracklet association with online target-specific metric learning. IEEE Conference IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015. p. 1234–241.
- Leal-Taixé L, Milan A, Reid I, Roth S, Schindler K. MOTChallenge 2015: Towards a benchmark for multi-target tracking [Internet]. 2015 Apr. Available from: https:// arxiv.org/abs/1504.01942.
- Wu Y, Lim J, Yang MH. Online object tracking: A benchmark. Conference on Computer Vision and Pattern Recognition Workshops; 2013. p. 2411–18.
- 13. Zhong W, Lu H, Yang M-H. Robust object tracking via sparsity-based collaborative model. IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2012.
- 14. Hare S, Saffari A, Torr PHS. Struck: structured output tracking with kernels. ICCV; 2011.
- 15. Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2003; 25(5):564–77.

- 16. Sevilla-Lara L, Learned-Miller E. Distribution fields for tracking. IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2012.
- 17. Mei X, Ling H. Robust visual tracking using L1 minimization. ICCV; 2009.
- Jia X, Lu H, Yang M-H. Visual tracking via adaptive structural local sparse appearance model. 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2012.
- Kazantsev P, Skribtsov P, Surikov S. Fast multi-object tracking-by-detection using tracker affinity matrix. Indian Journal of Science and Technology. 2016 Jul; 9(27). DOI: 10.17485/ijst/2016/v9i27/97696.
- 20. Matas JO, Urban CM, Pajdla T. Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing. 2002; 22:384–93. DOI: 10.1016/j.imavis.2004.02.006.
- Kazantsev P, Skribtsov P. Pedestrian detection in RGB-D data using deep autoencoders. *American Journal of Applied Sciences*. 2015; 12(11):847–56. DOI: 10.3844/ ajassp.2015.847.856.
- Skribtsov P, Kazantsev P, Dolgopolov A. Head-shoulder detection using deep autoencoders Indian Journal of Science and Technology. 2016; 9(42):1–9.

- 23. Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images; 2009.
- Skribtsov P, Kazantsev P. Multi-layer neural network auto encoders learning method, using regularization for invariant image recognition. Indian Journal of Science and Technology. 2016; 9(27). DOI: 10.17485/ijst/2016/ v9i27/97704.
- 25. Bilmes JA. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report TR-97-021. International Computer Science Institute and Computer Science Division, University of California at Berkeley; 1998 Apr.
- Cormen TH, Leiserson CE, Rivest RL, Stein C. Chapter
 21: Data structures for Disjoint Sets. Introduction to Algorithms (Second ed.), MIT Press; 2001. p. 498–524.
- 27. Drones [Internet]. [Cited 2016 Dec 02]. Available from: http://www.dji.com/products.
- Aerial drone videos by Photolab-gr [Internet]. [Cited 2016 Mar 01]. Available from: https://www.youtube.com/channel/UC80WcHE-SEthC4KQ7N82_Eg.
- 29. Bernardin K, Stiefelhagen R. Evaluating multiple object tracking performance: The Clear MOT Metrics. Journal of Image and Video Processing. 2008; (3):1–10.