VISUAL TRACKING APPLYING DEPTH SPATIOGRAM AND MULTI-FEATURE DATA

Niloufar Salehi Dastjerdi and M. Omair Ahmad

Department of Electrical and Computer Engineering,
Concordia University, Montreal, Quebec, Canada

ABSTRACT

Object tracking, in general, is a promising technology that can be utilized in a wide variety of applications. It is a challenging problem and its difficulties in tracking objects may fail when confronted with challenging scenarios such as similar background color, occlusion, illumination variation, or background clutter. A number of ongoing challenges still remain and an improvement on accuracy can be obtained with additional processing of information. Hence, utilizing depth information can potentially be exploited to boost the performance of traditional object tracking algorithms. Therefore, a large trend in this paper is to integrate depth data with other features in tracking to improve the performance of tracking algorithm and disambiguate occlusions and overcome other challenges such as illumination artifacts. For this, we use the basic idea of many trackers which consists of three main components of the reference model, i.e., object modeling, object detection and localization, and model updating. However, there are major improvements in our system. Our forth component, occlusion handling, utilizes the depth spatiogram of target and occluder to localize the target and occluder. The proposed research develops an efficient and robust way to keep tracking the object throughout video sequences in the presence of significant appearance variations and severe occlusions. The proposed method is evaluated on the Princeton RGBD tracking dataset and the obtained results demonstrate the effectiveness of the proposed method.

KEYWORDS

Visual Tracking, Depth Spatiogram, Multi-feature Data, Occlusion Handling

1. INTRODUCTION

Object tracking is one of the most important problems in computer vision and it is a key component of many applications ranging from robotics, automatic driving, intelligent surveillance, to augmented reality. Object tracking is a challenging problem and its difficulties can arise due to object appearance changes, illumination changes, occlusion, and so on. In the past few decades, most researchers have taken their efforts on color-image-based tracking methods [1,2,3], and have achieved a great progress. However, the intrinsic character of color image is that it is obtained by color camera at the cost of losing information by projecting 3D to 2D, which makes the features extracted from it easily influenced by the changes of illumination. Another challenge in tracking is efficient use of features through severe conditions such as partial or total occlusion or scale changes. So, in general, the most desirable property of applying appropriate features is to make the object easily distinguishable against non-targets in the feature space. From one pixel within a color image, the RGB color features can be naturally extracted. It is then not difficult to transform them into other color spaces or to gray levels. In addition, gradient and text features can also be extracted by considering the pixel within a local neighborhood. In [4] 13 different linear combinations of R, G, B pixel values were used to

approximate 3D RGB color space using a set of 1D histograms is cheaper. Incremental Discriminative Color Tracking(IDCT) tracker [5] is another RGB tracker which utilized a discriminative method to provide an incremental object color modeling to separate the object from surrounding background in each frame. In order to describe a region of pixels in a higher level, one popular way is to use a feature descriptor based on statistics, such as the histogram which has been widely used in many computer vision applications to represent the pixel feature distribution. Due to the simplicity, intensity histograms are widely used to represent objects for recognition and tracking. However, spatial information of object appearance is missing in this holistic representation, which makes it sensitive to noise as well as occlusion in tracking applications. So, authors in [6] introduced spatiogram which is basically a histogram with additional spatial information. A spatiogram computes not only information about the range of the function like histograms, but also information about the (spatial) domain. In [7], a fixed number of object parts are dynamically updated to account for appearance and shape changes. Moreover, when the object and background have similar color, or occlusion occurs, or tracker encounter with any other hard situations, a drift problem may happen. Therefore, a number of ongoing challenges still remain and how to reduce the impacts of these factors becomes a key issue in the tracking field. As a rule of thumb, an improvement on accuracy under specific conditions can be obtained with additional processing of information. Hence, utilizing depth information can potentially be exploited to boost the performance of traditional object tracking algorithms. Research into combining depth and color data for tracking is still in its infancy [8]. However, it has been demonstrated that state-of-the-art RGB tracking algorithms can be outperformed by approaches that fuse color and depth, for example [8-10]. A limited number of RGB-D tracking algorithms have been developed due to the novelty of this research domain. Song et al. [11] proposed several baseline RGB-D tracking methods, most of which are powerful. They trained a SVM classifier using the HOG features extracted from both color and depth frames to discriminate the target from the background. The target is detected in the entire image plane and the detection score for the candidate is adjusted by a large displacement optical flow tracker. In Wang et al. [12], the target region's color histogram and optical flow, was combined with the target's mean depth, to track its motion. Their work did not detail how their model was updated, and their results were reported on sequences not publicly available. The algorithm presented by Garcia et al. in [13] extended the condensation-based RGB tracker of Klein and Cremers [14] to incorporate depth data and predict the 3D spatial state of the particles in the condensation algorithm. Their boosting classifier was built from a pool of grayscale, color, and depth features, and in particular, the invariant gradient features of [14] were extended to depth data. Targeting more accurate performance of tracking algorithm, in this paper the depth data is integrated with other features and makes our main contributions as follows:

- Creating a depth mask to handle the problem of similar background color or texture in order to improve accuracy.

- Handling the occlusion state by computing depth spatiograms of target and occluder, which helps us to find the target position and recover the occlusion state.

The rest of the paper is organized as follows. Section 2 describes the details of the proposed method. The experimental results and performance evaluations are provided in Section 3. Finally, the conclusions are drawn and the future work is described in Section 4.

2. PROPOSED METHOD

With the added data from a depth sensor, tracking accuracy can be improved. The depth sensor effectively adds an additional channel to the image data, increasing the information density. Now an object can be described not only by its' color, but also by its' shape. In this paper, we integrate a combination of several image features including color and depth to track the object in order to

handle challenges such as similar background color, and occlusion. Such combination boosts the tracking performance and improves the accuracy. Figure 1 indicates the processing diagram of our tracker. As can be seen, it is comprised of four main components. Our motivation to use this diagram stems from the reference model for trackers which presented by Smeulders et al. [15].
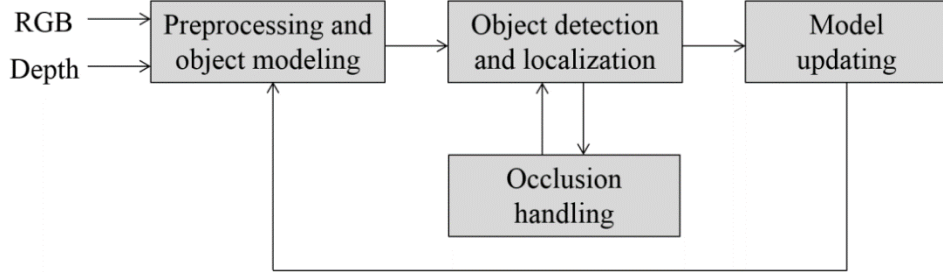

Figure 1. Processing diagram of proposed tracking method.

As the reference model does not consist of extra components, it provides the right balance between computational efficiency and accuracy. For this, we use the basic idea of IDCT tracker [5] which consists of three main components of the reference model, i.e., object modeling, object detection and localization, and model updating. However, there are key differences between our proposed system and IDCT tracker [5] which enables us to improve the performance of tracker. Our forth component, occlusion handling, utilizes the depth spatiogram of target and occluder to localize the target and occluder for handling occlusion. We develop an efficient and robust way to keep tracking the object throughout video sequences in the presence of significant appearance variations and severe occlusions. Complete description about our proposed system explained in the following subsections which provide more details about each component and highlight our motivations to use them.

2.1. Preprocessing and Object Modeling

The appearance of the target is represented by visual cues. The appearance model implies a constancy of a certain property to transfer one frame to the next. Without any such constancy assumption tracking cannot work. In the first step of our system, the bounding box is provided in the initial frame. However, the common argument against a bounding box is that background pixels may be confused with the target pixels. So we tend to the deliberate inclusion of some background in the description of an object to provide much information. So, using image features of object region and background region surrounding the object, an object model is created similar to [5, 16], to represent the target. Such a representation, although simple, is observed to be very powerful at detecting the foreground and the background [16]. Positive part of the log-likelihood ratio of the object and background determines the object model as in (1),

$$LR_{fb} = max\{ln\frac{max\{H_{obj}(fb),\varepsilon\}}{max\{H_{bg}(fb),\varepsilon\}},0\} \tag{1}$$

where $H_{obj}(fb)$ is the histogram computed within the object rectangle, and $H_{bg}(fb)$ is the histogram of background rectangle which is the region surrounding the object. Index fb indicates the feature bin and ranges from 1 to total number of histogram bins. ε is a small nonzero value to avoid dividing by zero. Note that, different from [5] which authors used red, green, and blue features of object and background regions, in this paper additional features are utilized to improve the efficiency of the model as in (2),

$$\mathbf{F}(x,y) = [R,G,B,H,S,I_x,I_y] \tag{2}$$

where $\mathbf{F}(x, y)$ represent feature vector. R, G, and B are red, green, and blue channels of image. H and S stand for hue and saturation features. The last two features represent the first order intensity gradients for the pixel (x, y). In parallel, the depth image is preprocessed to normalize the depth values between 0 and 255. The closer to the camera, the larger the depth value is. Inspiring from [6], depth spatiogram of bounding box is computed as follows,

$$DS_b = \langle n_b, \mu_b, \Sigma_b \rangle, \qquad b = 1, \dots, B$$

$$n_b = C \sum_{i=1}^{N} \delta_{ib} \tag{3}$$

Where N is the total pixel number of a target image. $\delta_{ib} = 1$ if pixel i is within the b-th bin, or else $\delta_{ib} = 0$. The normalising constant, C, ensures than all bins sum to one (i.e. $n_b = \sum_{i=1}^{B} n_i = 1$). For *2nd*-order spatiograms, the spatial mean, μ_b, and covariance, Σ_b, are computed as (4) and (5):

$$\mu_b = \frac{1}{\sum_{i=1}^{N} \delta_{ib}} \sum_{i=1}^{N} c_i \delta_{ib} \tag{4}$$

$$\Sigma_b = \frac{1}{\sum_{i=1}^{N} \delta_{ib}} \sum_{i=1}^{N} (c_i - \mu_b)(c_i - \mu_b)^T \delta_{ib} \tag{5}$$

where $c_i = (x_i, y_i)^T$ is the spatial position of pixel i. As the depth value within an object is continuous, there will be a rising peak that has the maximum in the depth histogram as shown in the Figure 2(d). Our assumption is that the bounding box of the target is without occlusion in the first frame, which means the target is the closest object to the camera. In addition, similar to [9] which introduced a fast tracker, a depth segmentation approach is done by applying K-means to estimate initial clusters or regions of interest (ROI), followed by, connected component analysis that is analyzed in the image plane to distinguish between objects located within the same depth plane. So we obtain connected component matrix (CCR) which is used in the object localization.

2.2. Object Detection and Localization

Similar to [5], we use the positive log-likelihood ratio LR, as a mapping function to provide an intermediate map (IM), from the object region. However, different from [5], additional features are applied in the image region to improve accuracy. So intermediate map is represented as in (6).

$$IM(x_i, y_j) = LR(\mathbf{F}(x_i, y_j)) \tag{6}$$

where (x_i, y_j) is pixel coordinate and $\mathbf{F}(x_i, y_j)$ shows feature vector, in the object region. Then, employing the depth data, we create a masked map as (7). This map is used for localizing the object in the next frame precisely and alleviating the problem of same color or texture in background which may lead to drift in tracking.

$$M(x_i, y_j) = IM(x_i, y_j) * CCR(x_k, y_m) \tag{7}$$

with $CCR(x_k, y_m)$ denotes the matrix of connected component with the same size of IM and includes zero values for coordinates containing non-targets. So, (x_k, y_m) indicates pixel coordinate which is non-zero and has same depth as target's depth. In order to localize the object, mean-shift algorithm is applied on masked map which obtained from (7). The centroid of the masked map of the object in the current frame is used to localize the object in the next frame. At

each iteration, the center of the object rectangle is shifted to the center of the current masked map of the object as (8) and (9). The iteration will repeated until the object is placed inside the rectangle completely. Similar to [5], the maximum number of mean-shift iterations is set to 3 and complete convergence is considered for the centroid movement less than 2 pixels.

$$x_{new} = \frac{\sum_{i=1}^{N} M_i \times x_i}{\sum_{i=1}^{N} M_i} \tag{8}$$

$$y_{new} = \frac{\sum_{j=1}^{N} M_j \times y_j}{\sum_{j=1}^{N} M_j} \tag{9}$$

2.3. Model Updating

To update the target model to the recent observations, positive log-likelihood ratio at current frame (LR^t) is used similar to [5]. Once the object location at the current frame is computed by the mean-shift, LR^t is applied to update the previous object model, LR^{t-1}. So, the updated object model, LR^{t+1}, is computed as (10),

$$LR^{t+1} = \rho \times LR^t + (1 - \rho) \times LR^{t-1} \tag{10}$$

ρ is a forgetting factor to make the balance between the old and new observations and is set to 0.1.

2.4. Occlusion Handling

For each incoming frame, after obtaining the object location using the processing steps explained in previous subsections, the depth histogram inside the object bounding box is computed to decide whether occlusion occurs or not. Figure 2 shows the object inside BB without occlusion, with partial occlusion, with total occlusion, and their corresponding depth histogram, respectively. As can be seen, occlusion occurs when a new rising peak with a smaller depth value of the target, appears. So, when occlusion occurs, the depth value of the occluder will be smaller than that of the target. Occlusion state happens when the smaller depth value, d, meets the following condition:

$$d < |\mu - 2\sigma| \tag{11}$$

where μ and σ are the mean value and standard deviation of the Gaussian distribution of the depth histogram respectively.
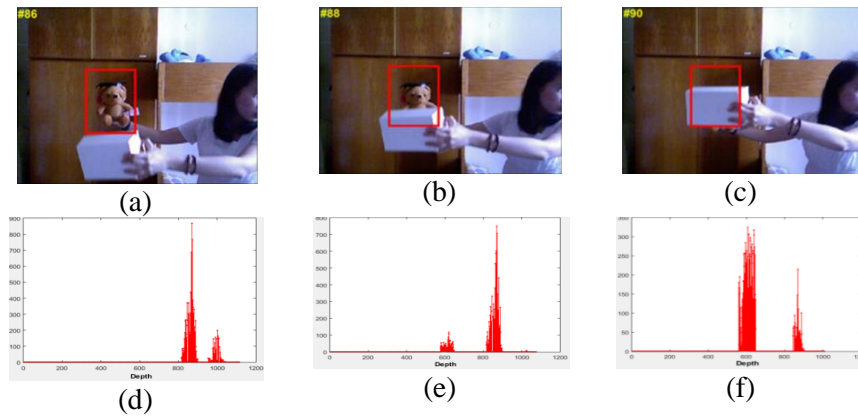

Figure 2. (a) target only, (b) target with partial occlusion, (c) target with total occlusion, and (d-f) corresponding depth histogram inside the red rectangle, respectively.

To recover from the occlusion state at each occluded frame, depth spatiogram of the region inside bounding box is analyzed. Therefore, in order to locate the occluder's position in the current bounding box, we compute the depth spatiogram to obtain the depth value and depth location of the occluder. It should be noted that the use of the image coordinates with real-world depth facilitates defining the search space and yields a more accurate estimate of the position of the occluded target. Figure 3 shows the preliminary result of applying depth spatiogram to obtain μ_x and μ_y of target and occluder. Figure 3 (a) and Figure 3 (b) show the RGB and depth data of BB in frame 40 of sequence 'bear_front'. Figure 3 (c) indicates the obtained center locations of occluder and target. We obtained the coordinates (-0.2024, -0.4437) for the center of target, and (0.0223, 0.5166) for the center of occluder, in form of (x, y).

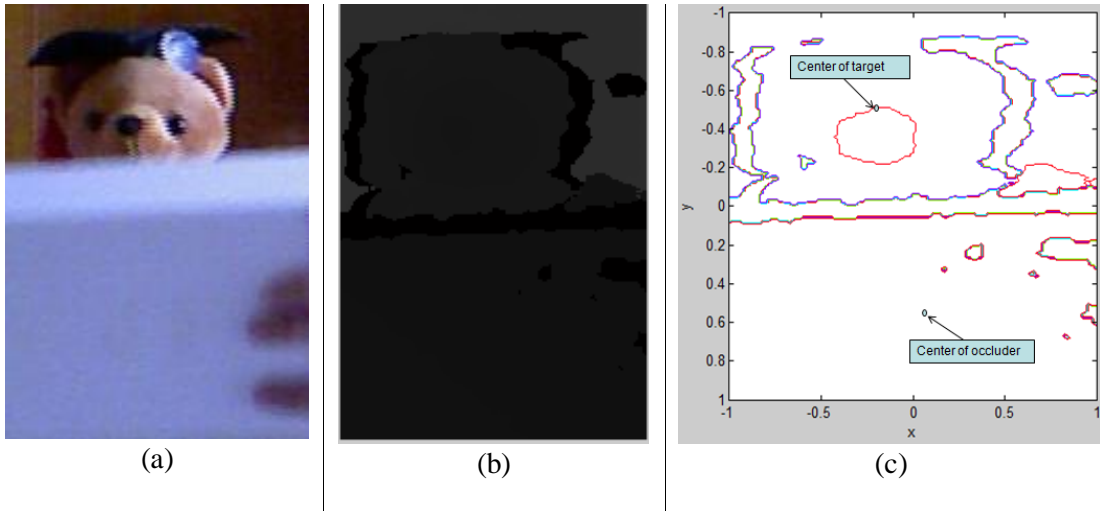| (a) | (b) | (c) |

Figure 3. (a) RGB data, (b) Depth data, (c) center location of target and occluder, in the BB of frame 40 from 'bear' sequence.

Suppose the target is detected near-fully occluded in the current frame t. First, a search region is defined at the centroid of occluder. Then, the area around the occluder is searched to locate the target in the next frame $t+1$. The object has a newly rising peak with a depth value smaller than the depth value of the occluder. So according to the spatial information obtained from depth spatiogram, the object is highly likely to re-emerge in those image areas close to the center of obtained spatial information. We then create a new BB centered at the obtained spatial means, and covering the obtained depth around the center, as object candidate. So the size of the new BB is correspondence to the number of pixels whose value is that of the obtained depth. For the objectcandidate, its similarity, S, is compared with object in the previous frame using L1-norm as,

$$Dist = \|\mathbf{F}^{t-1} - \mathbf{F}'^{t}\|_1$$
$$S = |1 - Dist| \tag{12}$$

where $\mathbf{F}^{t-1}$ denotes feature vector of object in previous frame and $\mathbf{F}'^{t}$ indicates the feature vector of object candidate in current frame. Note that L1-norm is applied here because it is simple and effective. If the similarity of the candidate is greater than a given threshold (90% in our experiments), it is regarded as object. Otherwise, we expand the size of candidate's BB to create a search area, and search for a region having most similar features as target. The search is performed by sliding-window from left to right and from top to bottom in the search area. The window size is fixed and is equal to the size of candidate's BB before expanding. The search window jumped horizontally 10% of the width or vertically 10% of the height of the search area. Then, the best 10% matchings are extracted as our candidates, and the dissimilarity of the object

and all candidates are computed. Finally, the region with the smallest distance is selected as the best candidate. This candidate is passed to the second step, i.e., object detection and localization, and the algorithm will be continued to track the object throughout the sequence.

3. EXPERIMENTAL RESULTS

We evaluate our method Princeton Tracking RGBD dataset [11] which recorded with Microsoft Kinect v1. It contains 95 test sequences with different kind of challenges such as complex background clutter, various target types and target sizes, partial and complete occlusion, fast movement, shape deformation and distortion, different scene types and so on. Besides, we also compare the performance of the proposed method with a number of tracking methods, including mean-shift tracker with Corrected Background Weighted Histogram (CBWH)[17], Locality Sensitive Histogram Tracker (LSHT)[18], and IDCT tracker[5]. CBWH reduce background's interference in object localization by computing a color histogram with new weights to pixels in the target candidate region. LSHT computes a locality sensitive histogram that was computed at each pixel location along with a floating-point value corresponds with each bin to save the occurrence of an intensity value.

3.1. Objective Results

In order to evaluate the performance of our proposed method, two objective measures are used. Average center location error and the average overlap rate. The average center location error (ACLE) [19] is a widely used metric that computes the average Euclidean distance according to (13),

$$ACLE = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(X_i - X_i^g)^2 + (Y_i - Y_i^g)^2} \tag{13}$$

where $[X_i, Y_i]$ denotes the center location of the object obtained by the tracker which is determined by the central point of the object rectangle. $[X_i, Y_i]$ indicates the center of the ground truth rectangle. n is the total number of frames and i ranges from 1 to n. The average overlap ratio (AOR) measures the overlap ratio between the estimated BB predicted from the tracker (B_t) and the annotated BB (B_t^g) according to (14),

$$AOR = \frac{B_t \cap B_t^g}{B_t \cup B_t^g} \tag{14}$$

The results evaluated by the above measures are shown in Table 1. As can be seen, our method achieves the best or the second best performance when compared with other tracking methods.

Table 1. The average center location errors (ACLE) and average overlap rate (AOR) of the evaluated methods on the sequences.

Algorithm	Seq.	bear_front	child_no1	face_occ5	New_ex_occ4	zcup_move_1
CBWH	ACLE	20.4	31.6	18.2	25.6	18.4
	AOR	0.59	0.52	0.60	0.75	0.68
LSHT	ACLE	26.1	35.0	21.4	26.0	17.1
	AOR	0.75	0.49	0.61	0.78	0.81
IDCT	ACLE	12.3	16.9	25.0	27.2	**6.7**
	AOR	0.67	**0.72**	0.78	0.63	**0.89**
Ours	ACLE	**4.5**	**9.9**	**10.8**	**19.7**	16.4
	AOR	**0.82**	0.66	**0.92**	**0.84**	0.52

3.2. Subjective Results

Subjective results of our method on three challenging sequences from [11] are shown in Figures 4-6. The red rectangle shows the tracked object. Figure 4 consists of frames from 'bear_front' sequence. The challenges in this sequence include similar background and several encounters of occlusions. Figure 5 indicates a number of frames from 'zcup_move_1' sequence with challenges like similar background. Figure 6 indicates frames from 'new_ex_occ4' sequence which consist of challenges such as cluttered background, dark illumination and occlusion.

Figure 4. Tracking results for the 'bear_front' sequence.

Figure 5. Tracking results for the 'zcup_move_1' sequence.

Figure 6. Tracking results for the 'new_ex_occ4' sequence.

4. CONCLUSION

Object tracking is one of the most important problems in computer vision and its goal is accurately locate a moving target in a video sequence and discriminate target from non-targets in the feature space of the sequence. For tracking module, we first selected the main components of common tracking frameworks. Then, by applying depth data, we improved the performance of object tracking. To this, we created a masked map to overcome the problem of similar background color or texture. We also computed the depth spatiogram to handle the occlusion and better recovering of occlusion state, by finding the position of target and occluder. The experimental results show that depth features indeed help to improve the tracking result. This is because the depth map is not easily influenced by the change of illumination, and the depth cue is more stable and indicative for object localization. Our proposed methods not only achieve more accurate results in the experiments, but also is efficient as it does not require complicated pre-processing or any training, which will facilitate practical real-time tracking applications. Future work should be aimed at efficient selection of features. Different feature combinations generally produce different detection performances and previous works generally reported better results using more features. So, how to select proper feature set for detecting a specific object to ensure good performance in terms of detection accuracy can be considered in further work.

REFERENCES

[1]   Liu, Gaocheng, et al. (2018). "Object tracking in vary lighting conditions for fog based intelligent surveillance of public spaces", *IEEE Access*, *6*, pp29283-29296.

[2]   Crivelli, Tomas Enrique, Juan Manuel PEREZ RUA, and Patrick Perez, (2018) "Method and apparatus for object tracking and segmentation via background tracking", U.S. Patent Application 15/314,497.

[3]   Guo, Siqiu, et al., (2018) "Color Feature-Based Object Tracking through Particle Swarm Optimization with Improved Inertia Weight", *Sensors*, Vol. 18, NO. 4, p1292.

[4]   Wang, Dong, et al., (2013) "Fast and effective color-based object tracking by boosted color distribution," *Pattern Analysis and Applications*, Vol. 16, NO. 4, pp647-661.

[5]   Asvadi, Alireza, et al., (2014) "Online Visual Object Tracking Using Incremental Discriminative Color Learning", *CSI Journal on Computer Science and Engineering (JCSE),* Vol. 12, No. 2, pp16-28.

[6]   Birchfield, Stanley T., and Sriram Rangarajan, (2005) "Spatiograms versus Histograms for Region-Based Tracking" *Computer Vision and Pattern Recognition (CVPR)*, San Diego.

[7]   Čehovin, Luka, Matej Kristan, and Alesš Leonardis, (2011) "An adaptive coupled-layer visual model for robust visual tracking*", International Conference on Computer Vision (ICCV).*

[8]   Camplani, Massimo, et al., (2017) "Multiple human tracking from RGB-D data: a survey", *IET Comput. Vis.*

[9]   Camplani, Massimo, et al., (2015) "Real-time RGB-D tracking with depthscaling kernelised correlation filters and occlusion handling", *BMVC*, pp145.1–145.11.

[10]   García, Germán Martín, et al., (2012) "Adaptive multi-cue 3D tracking of arbitrary objects", Pattern*Recognition*, pp357–366.

[11]   Song, Shuran, and Jianxiong Xiao, (2013) "Tracking revisited using RGB-D camera: unified benchmark and baselines", *IEEE International Conference on Computer Vision (ICCV)*, pp233–240.

[12]   Wang, Qi, Jianwu Fang, and Yuan Yuan, (2014) "Multi-cue based tracking", *Neurocomputing*, Vol. 131, NO. 0, pp227 – 236.

[13]   García, Germán Martín, et al., (2012) "Adaptive multi-cue 3D tracking of arbitrary objects", Pattern *Recognition*, pp357–366.

[14]   Klein, Dominik A., and Armin B. Cremers, (2011) "Boosting scalable gradient features for adaptive real-time Tracking", *ICRA*, pp4411–4416.

[15]   Smeulders, Arnold WM, et al., (2014) "Visual tracking: An experimental survey" IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 36, NO.7, pp1442-68.

[16]   Fan, Jialue, Xiaohui Shen, and Ying Wu, (2012) "Scribble tracker: a matting-based approach for robust tracking" *IEEE Transactions on Pattern Analysis and Machine Intelligence* VOL. 34, NO. 8, pp1633-1644.

[17]   Ning, Jifeng, et al., (2012) "Robust mean-shift tracking with corrected background weighted histogram", *IET Computer Vision*, VOL. 6, NO. 1, pp62-69.

[18]   He, Shengfeng, et al., (2013) "Visual tracking via locality sensitive histograms", Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, pp2427-2434.

[19]   Babenko, Boris, Ming-Hsuan Yang, and Serge Belongie, (2011) "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, VOL. 33, NO. 8, pp1619-1632.

AUTHORS

Niloufar Salehi Dastjerdi received the M.Sc. degree in computer science in 2012 from University Technology Malaysia. She is currently working toward the Ph.D. degree in the department of Electrical and Computer Engineering, Concordia University, Montreal, Canada. Her research interests include visual tracking, computer vision, image processing, and pattern recognition.

M. Omair Ahmad (S'69M'78–SM'83–F'01) received the B.Eng. degree from Sir George Williams University, Montreal, QC, Canada, and the Ph.D. degree from Concordia University, Montreal, QC, Canada, both in electrical engineering. From 1978 to 1979, he was a Faculty Member with the New York University College, Buffalo, NY, USA. In September 1979, he joined the Faculty of Concordia University as an Assistant Professor of computer science. He joined the Department of Electrical and Computer Engineering, Concordia University, where he was the Chair with the department from June 2002 to May 205 and is currently a Professor. He holds the Concordia University Research Chair (Tier I) in Multimedia Signal Processing. He has authored or coauthored extensively in the area of signal processing and holds four patents. His current research interests include the areas of multidimensional filter design, speech, image and video processing, nonlinear signal processing, communication DSP, artificial neural networks, and VLSI circuits for signal processing. He was a Founding Researcher at Micronet, Ottawa, from its inception in 1990 as a Canadian Network of Centers of Excellence until its expiration in 2004. Previously, he was an Examiner of the order of Engineers of Quebec. He was an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS PART I: FUNDAMENTAL THEORY AND APPLICATIONS from June 1999 to December 2001. He was the Local Arrangements Chairman of the 1984 IEEE International Symposium on Circuits and Systems. In 1988, he was a member of the Admission and Advancement Committee of the IEEE. He has served as the Program Co-Chair for the 1995 IEEE International Conference on Neural Networks and Signal Processing, the 2003 IEEE International Conference on Neural Networks and Signal Processing, and the 2004 IEEE International Midwest Symposium on Circuits and Systems. He was a General Co-Chair for the 2008 IEEE International Conference on Neural Networks and Signal Processing. He is the Chair of the Montreal Chapter IEEE Circuits and Systems Society. He is a recipient of numerous honors and awards, including the Wighton Fellowship from the Sandford Fleming Foundation, an induction to Provosts Circle of Distinction for Career Achievements, and the Award of Excellence in Doctoral Supervision from the Faculty of Engineering and Computer Science of Concordia University.