2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Dynamic visual localization and tracking method based on RGB-D information

Chunxia Yin, Cai Luo, Yipeng Li, and Qionghai Dai
Broadband Networks & Digital Media Lab
Department of Automation
Tsinghua University
Beijing, China
(cxyin, luo_cai, liep, qhdai)@tsinghua.edu.cn

***Abstract*—This paper proposes a new dynamic visual localization and tracking method using RGB-D camera. The proposed method makes use of band-width matrix, combines particle filter and mean shift with a new strategy, avoids falling into local optimum, and maintains particle diversity with only a few sampled points. A fast object searching strategy is successfully used to find the missing object. Experiments show that the proposed method runs robustly in complex scenes. The object 3-D parameters relative to the camera center can be estimated with a RGB-D camera, and that makes significant sense in spatial localization.**

I. INTRODUCTION

Object tracking is a challenging problem in image analysis, and is becoming more and more important in context-understanding, human-computer interaction, security and surveillance, traffic control, and so on. Numerous algorithms have been proposed in recent years, but robust tracking remains a difficult task due to many factors like illumination, occlusion, pose angle, camera motion, among others.

Tracking methods based on mean shift are known as simple and robust, but prone to get into local optimum. Comaniciu is the founder of this branch [1], [2], researchers after him have done much extending work in many aspects [3], [4]. Particle filter is a probabilistic algorithm based on state estimation and is outstanding in non-linear and non-Gaussian system [5]. In recent years, many methods on combination of mean shift and all kinds of filters appear [6].

With development of the supervised tracking methods, many new algorithms have been proposed, like MIL [7], L1 [8], TLD [9], and CT [10]. Paper [11] gives a summary and analysis of these methods. But there are still two important problems that block these algorithms to be really robust: occlusion and scale changing. In this paper, we focus on occlusion and scale tracking problem.

With adaptive-bandwidth function, we propose a new tracking method combines the advantages of particle filter and mean shift, it is a real-time approach to process occlusion and scale problem. A fast searching strategy is proposed to re-pick the missing object.

All the algorithms mentioned above are 2-D tracking methods. In technical application, the relative spatial location between the object center and the camera center is more significant. In [12] the altitude is achieved using the barometer measurements, sometimes altimeter such as laser range finder is also being used. These methods are non-vision and are confined to ground object. Binocular stereo vision system can get the depth information, but it depends too much on the object feature and is hard to get good result in practical application, especially in dynamic systems. Here we use a RGB-D camera to get the space relative position between the object and the camera center. Using depth cameras is not conceptually new, equipment such as Kinect has made such sensors accessible to all. The quality of the depth sensing, given the low-cost and real-time nature of the device, is compelling, and has made the sensor instantly popular with researchers and enthusiasts [13].

In section II, the bandwidth matrix is introduced, which is used to track object size adaptively. In section III, the theory of mean shift and particle filter is reviewed. In section IV, the proposed method is discussed and a fast local searching strategy is introduced. In section V, Proposed method is tested, the experiments results are shown in section V and get the spatial position of the tracking object using a RGB-D camera Microsoft Kinect. A conclusion is given in section VI.

This work was supported by the Project of NSFC (No. 61035002 & 61271450), funded by Beijing Key Laboratory of Multi-dimension & Multi-scale Computational Photography (MMCP), Tsinghua University.

II. Bandwidth Matrix

We define an ellipse area to describe the tracked target:

$$S = \{s|(s - X_0)^T H^{-1}(s - X_0) < \sigma^2\} \quad (1)$$

X_0 is the center of S, s is the pixel in S, ϕ is the rotation angle, a and b are variables associated to an elliptical shape. σ is a factor determined by density kernel function. In this paper we use Gaussian kernel:$K(X) = c^{-\frac{X}{2}}$, and $\sigma = 2.5$. σa represents the long half shaft length and $\sigma b$the short shaft. Size and direction of S are totally related to σ and H.

From reference [4] we know, H can be defined as a positive definite symmetric matrix.

$$H = AA^T = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \quad (2)$$

where

$$A = R(\phi) \cdot \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \quad (3)$$

$$R(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \quad (4)$$

From (2) to (4) we can get

$$\begin{aligned} a &= \sqrt{\frac{1}{2}\left[h_{11} + h_{22} + \sqrt{4h_{12}^2 + (h_{11} - h_{12})^2}\right]} \\ b &= \sqrt{\frac{1}{2}\left[h_{11} + h_{22} - \sqrt{4h_{11}^2 + (h_{11} - h_{12})^2}\right]} \\ \phi &= \frac{1}{2} atan2(2h_{12}, h_{11} - h_{22}) \end{aligned} \quad (5)$$

The objective is to find x and H in order to make $\hat{P}_u$ in (10) to reach its maximum. The optimum bandwidth matrix is denoted as:

$$H_g = \frac{\sum_{s \in S} \omega(s)(X - s)(X - s)^T}{\sum_{s \in S} \omega(s)} \quad (6)$$

where is $\omega(s)$ is the weight of pixel s.

III. Mean Shift and Particle Filter

A. Mean Shift Algorithm

The theory of mean shift and its realization have been described explicitly in paper [1] and [2]. Here we introduce the functions we will use later in our new tracking method. The computational module is based on mean shift iterations, which finds the most probable target position in the current frame. The similarity between object model and the target candidate is expressed by a metric deduced from the Bhattacharyya coefficient.

1) Color representation: The feature distribution in object model is initialized as $\{\hat{q}_u\}_{u=1,\ldots m}$,

$$\hat{q}_u = C \sum_{s \in S_0} K[(X_0 - s)^T H_0^{-1}(X_0 - s)]\delta(b(s) - u) \quad (7)$$

where u is the bin index in histogram; S_0 is the initial elliptic area of s; C is a normalization factor,

$$C = \frac{1}{\sum_{s \in S_0} K[(X_0 - s)^T H_0^{-1}(X_0 - s)]} \quad (8)$$

$b(s)$ maps pixel s to the feature space histogram.

The mean shift vector is denoted as:

$$m(X) - X = \frac{\sum_{s \in S} G_H(X - s)\omega(s)s}{\sum_{s \in S} G_H(X - s)s} - X \quad (9)$$

where $G_H = |H|^{-\frac{1}{2}} G(X^T H^{-1} X)$, $G(x) = -K'(x)$. The feature distribution in target candidate model is $\{\hat{p}_u\}_{u=1,\ldots m}$

$$\begin{aligned} &\hat{p}_u(X_1) = \\ &C_H \sum_{s \in S} |H|^{-\frac{1}{2}} K[(X_1 - s)^T H^{-1}(X_1 - s)]\delta[b(s) - u] \end{aligned} \quad (10)$$

where C_H is the normalization factor, X_1 is center of the target candidate, $\omega(s)_{s \in S_0}$ is the weight value,

$$\omega(s) = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{p}_u(x_0)}}\delta(b(s) - u) \quad (11)$$

2) Distance minimization: The Bhattachayya coefficient is given by:

$$\rho(\hat{p}(X_1), \hat{q}) = \sum_{u=1}^{m} \sqrt{\hat{p}_u(X_1)\hat{q}_u} \quad (12)$$

The algorithm aims at finding the location X_1 and bandwidth H to make $\rho(\hat{p}(X_1), \hat{q})$ achieve its maximum value. This is also the action that leads to the problem of local optimum.

B. Particle Filter Algorithm(PF)

We sample particles randomly around the object center, detect the most probable object and transfer particles through a transition function $p(x_k|x_{k-1})$, which is a second order auto-regressive function in our experiment:

$$x_k = Ax_{k-2} + Bx_{k-1} + v \quad (13)$$

where A and B are coefficient of the AR function, v is a random number obeying standard normal distribution. We use a joint resampling method with mean shift

to keep particle diversity. x_k and y_k represent the particle state and observation value respectively at time k. Particle weight can be calculated by

$$w_k = w_{k-1} p(y_k|x_k) \tag{14}$$

where $p(y_k|x_k)$ is the plausibility function.

If the mean shift process detects the object correctly, then object scale is decided by the optimum bandwidth matrix above. If mean shift is out of function, the object scale can be predicted by the transition function.

IV. Adaptive Bandwidth Mean Shift and Particle Filter Algorithm (ABMSPF)

ABMS method in paper [4] can track the size of object automatically, but it's prone to fall into local optimum, and is sensitive to changing illumination. It is incapable in complex environment such as serious occlusions or clutter of similar colors. Particle filter method doesnt need to extract features and can process negative information, avoid being trapped in local optima. But localization with particle filter is not precise. It's time-cost and leads to a decreasing diversity as time runs out.

We propose the ABMSPF method, which avoids the shortcomings of both mean shift and the particle filter. It uses mean shift as the main frame. The optimal bandwidth matrix is used to track the object size, motion direction and object location adaptively. Particles are used as assisting tools to estimate the possible location, keeping the object from being kidnapped by similarities, localizing the object when serious occlusions or totally shielding occurs. Besides, the proposed tracking algorithm is capable of recovering itself quickly from failure. The innovative principal strategies in ABMSPF are described as follows.

A. Resample Particles at the Newly Detected Object Location

Particles are sampled at the center of the ellipse target area detected by mean shift. When the similarity ζ between object model and target candidate exceeds a threshold th, here $th = 0.85$, all the particles are re-sampled at the center of the object area. Then the particle filter algorithm begins to operate. This procedure gathers the particles at each time when the object is precisely detected. It keeps the diversity with only a few particles, reduces computational complexity, and improves tracking performances under occlusions or clutters.

B. Fusion of Different Location Information

Given that the object location achieved by iterative optimization in mean shift is X_{ms}, and the estimated location with particles is X_{pf}, $|X_{ms}X_{pf}|$ is the distance between X_{ms} and X_{pf} . Smaller distance means more precise localization. If $X_{ms}X_{pf}$ is big enough, that always means mean shift falls into local optimum, or a "false object" is tracked, or big error occurs with particles estimation. All the situations above will leads to a failure in tracking. In most cases, information fusion helps to find the right object and stops a possible crisis stated above, and X_c is the fusion result.

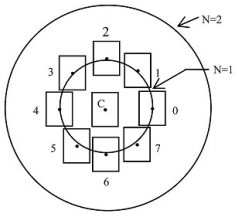
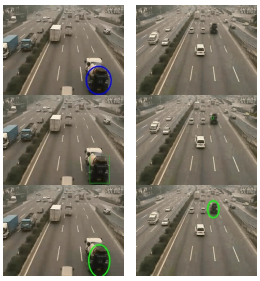


Fig. 1: Diagram for local searching strategy



(a) No. 1 (b) No. 64

Fig. 2: Tracking under scale changing

The information fusion principle is described as follows:
If $\zeta \geq th$, a and σ are parameters in object ellipse:

$$X_c = \begin{cases} \zeta \cdot X_{ms} + (1-\zeta) \cdot X_{pf} & if|X_{ms}X_{pf}| < \sigma \\ X_{ms}, & if\sigma \leq |X_{ms}X_{pf}| \leq \sigma a \\ Null, & if|X_{ms}X_{pf}| > \sigma a \end{cases} \tag{15}$$

if $r < \zeta < th$, (in our experiment $r = 0.5$), the state of object has changed or occlusion appears:

$$X_c = \begin{cases} X_{pf}, & if|X_{ms}X_{pf}| < \sigma a \\ Null, & if|X_{ms}X_{pf}| > \sigma a \end{cases} \tag{16}$$

if $0 \leq \zeta \leq r$, object is lost or totally shielded, tracking relies on the estimation of particles:

$$X_c = X_{pf} \tag{17}$$

C. Fast Searching Strategy

Once the object is detected as lost, it's necessary to find the object if it appears again in frames. Searching methods such as blob detecting or traversing is based on global searching strategy, which is time consuming. Assume that at the frame before the object is lost, the object center is C, the motion angle is $\phi, \phi \in [-180°, +180°]$, the external rectangle of the object is determined by 'height' and 'width'. According to motion continuity, the most possible location is around C near the direction ϕ. The searching strategy is shown in Fig.1, N is searching layers while d is the searching radius, there are $8*N$ searching directions at each layer and the starting direction is $n, n = (0, 1, \ldots, 8*N)$. The searching order at each layer is: (n, (n+1)mod(8*N), (n-1)mod(8*N), (n+2)mod(8*N), ...)

where

$$d \in ((N+1) * width, (N+1) * height) \tag{18}$$

$$n = floor(\frac{angle}{360/(N*8)}, n \subset \{0, 1, \cdots, 7\}) \tag{19}$$

$$angle = \begin{cases} \theta, & if(\theta \geq 0) \\ -\theta, & if(\theta < 0) \end{cases} \tag{20}$$

For example, if $N = 1$, then searching order in Fig.1 is $(1, 2, 0, 3, 7, 4, 6, 5)$. Usually the object can be found at the first layer, the local searching strategy is much faster.

If local searching fails a global searching strategy will begin, which detects the salient blobs first, then matches each blob to find the candidate object.

D. Implementation Steps

Applying the optimal bandwidth matrix and searching strategy above into object detection, we combine the estimation function of particles with mean shift, to form the ABMSPF algorithm. Implementation steps of the algorithm are stated in Table I.

V. Experiment

In experiment, we compare the proposed algorithm ABMSPF with ABMS and PF. Videos are provided by http://www.ces.clemson.edu/∼stb/research/adafrag/, both the scene and the camera are moving.

The experiment is performed on a personal computer with 32bit operating system, Pentium 2.1G processor, RAM 2.0GB. The software is finished with Visual Studio 2010 and OpenCV 2.3.1.

(a) No.55 (b) NO.134 (c) No.194 (d) No.265 (e) No.277

Fig. 3: Tracking under changing illumination and partial occlusion

A. Tracking under Scale Changings and Occlusions

In Fig.2, the first line is the tracking result of ABMS, object is lost at frame 64. The second line is PF method, it is hard for PF method to track object scale adaptively and it lost object after frame50. The ABMSPF method in the third line tracks the scale changing object robustly.

From the first line to the third line in Fig.3 are the tracking results of ABMS, PF, ABMSPF respectively. There are different extent of occlusions and illumination changes in the scene. ABMS tracks precisely in the first 58 frames, the object lost at frame 59 for obvious illumination change. PF uses 200 particles in total and particles begin to diverge at frame 134. ABMSPF tracks well through the whole sequence. At frame 194 when serious occlusion occurs, the proposed method can still estimate object location. The number of particle used in ABMSPF is 30. Experiment shows that the method can work normally even if the particles are reduced to a number of 10.

In ABMSPF method, tracking gets into searching process (Fig.4). At frame 195 and frame 221, the object is found at the second layer as in Fig.1.

In Fig.5, the object walks into total occlusions and re-appears. PF (Fig.5 line2) method diverges at frame 91. ABMS method (Fig.5 line1) fails when the person is seriously occluded. At frame 136, when the person is absolutely shielded by the tree, ABMSPF is the only algorithm that can estimate the person's location rationally. At frame 145, the person walks through behind the tree and appears in the scene again, ABMS and our method re-pick the object, but PF fails for lack of an efficient re-searching function.

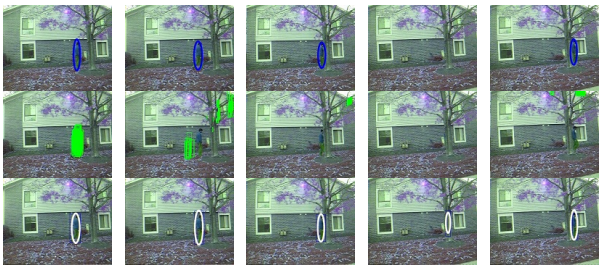
Fig.6 is another complex scenario with occlusions by similar objects. The ABMS method failed from the beginning (Fig.6 line1). The PF method was said to have a good usage on this video, but our experiment shows that PF method fails at many frames (Fig.6, line2.) and the ABMSPF method (Fig.6 line3) may be more satisfactory.

TABLE I: The ABMSPF Algorithm

	The ABMSPF Algorithm
1:	Initialization: select the object area S_0 manually, calculate bandwidth matrix H_0, object model $\{\hat{q}_u\}_{u=1,\ldots,m}$, weight $\{\omega(s)\}_{s\in S_0}$.
2:	Sample N particles in S_0, initialize particle weight as:$w^i = \frac{1}{N}$.
3:	Use (9) to shift the position X_0 to X_1, $X_1 = m(X_0)$, update S_0 to be $S_1^{'}$, calculate $\{\hat{q}_u\}_{u=1,\ldots,m}$, $\zeta = \rho(\hat{p}, \hat{q})$.
4:	Calculate optimal matrix H_1 by (6) in area $S_1^{'}$, then update $S_1^{'}$ to be S_1 according to the optimal bandwidth matrix $H_{1]}$.
5:	If: $\parallel X_0 - X_1 \parallel < \varepsilon$ and $S_0 = S_1$, go to step 6 else: $X_0 \leftarrow X_1, S_0 \leftarrow S_1, H_0 \leftarrow H_1$, go to step 3.
6:	Calculate ζ, $X_{ms} = X_1$; get estimated object center X_{pf} by particle filter.
7:	Calculate the object center X_c at current frame by (15), (16) and (17).
8:	If: $X = Null$ or $0 \leq \zeta \leq r$, go to step 9 else: go to step 11.
9:	Re-search object using the fast searching strategy in sectione IV part C.
10:	If: object is found, calculate the new object parameters $S_0, H_0, \{\omega(s)\}_{s\in S_0}$, go to step 2. else: go to step 9.
11:	$S_0 = S_1, H_0 = H_1$, update $\{\omega(s)\}_{s\in S_0}$ according to (11).
12:	If: $\zeta \geq th$, go to step 2, else: update particles by transition function (13), update particle weight by (14), go to step 3.

(a) No.194 (b) No.195 (c) No.221

Fig. 4: Re-searching process in tracking



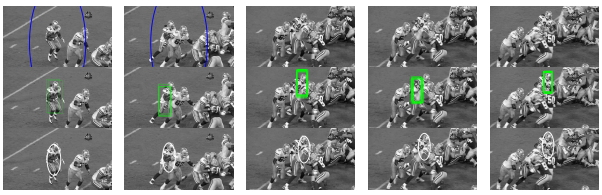
(a) No.84 (b) No.91 (c) No.120 (d) No.136 (e) No.145

Fig. 5: Tracking under complete occlusion

ABMSPF is proved to be viable in occlusions and scale changing. It tracks the object correctly even in complex environment. ABMS fails when illumination changes or the object is occluded to a severe extent, but the blob-detecting function can help it to re-pick the object when it appears again in the scene. Next we will discuss the location precision and computing efficiency.

B. Precision and Efficiency

In this section we select the scenes "Walk1" provided by http://homepages.inf.ed.ac.uk/rbf/CAVIAR/, it provides the truth-value for analyzing the control precision of our algorithm.



(a) No.26 (b) No.52 (c) No.76 (d) No.79 (e) No.90

Fig. 6: Tracking under similar object occlusions

1) Control Precision Analysis: Fig.7 shows the tracking trajectory of the object center at each frame. The measured location is truth-value provided by the test library and another is achieved by tracking algorithm. Its apparent that the location precision in ABMSPF is best in the whole test. ABMS is precise when the object can be detected and tracked correctly, but when the object is lost, big error occurs, which affects system stability seriously.

2) Computing Time Analysis: ABMS needs about 16ms to process one frame if the object is not lost. But when the object is lost, system goes into blob searching stage and the time cost increases dramatically. Time cost in PF algorithm is about hundreds of milliseconds, it decreases to about 50ms with time passes, which may be due to the decreased particle diversity. ABMSPF uses the local searching strategy provided in this paper and the time cost keeps around 50ms-75ms without algorithm optimization, which is enough for real-time tracking task.

C. Kinect Application

We apply the tracking method with a Kinect [14] sensor. This tracking system can be used on platform of robotics or UAVs. With a Kinect, we can get not only the object image position, but also the space 3D

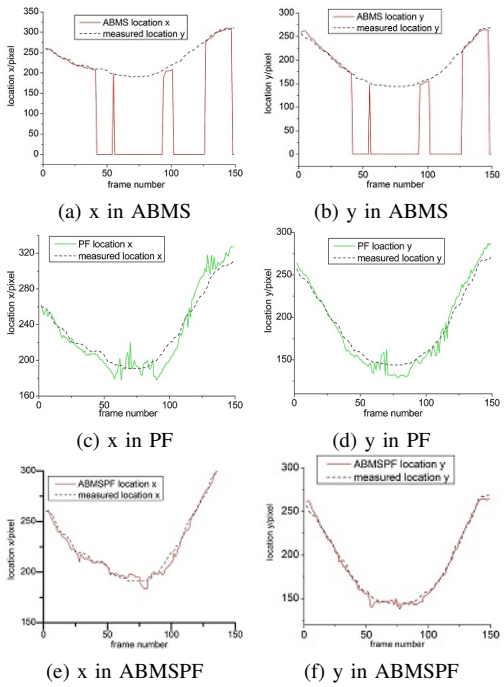


Fig. 7: Tracking trajectory and tracking error

position. Table II shows the Kinect tracking data in a test. Location results are given directly, representing space relative position between the object center and the camera optic center, the unit is mm. Negative data in X means the object is on the left side of optic axis.

VI. Conclusion

This paper proposes a new tracking method – ABMSPF method, and provides an efficient searching strategy to search the object quickly once it is lost. At last, the object 3-D parameters are achieved with a RGB-D camera. The proposed method combines advantages of ABMS and PF but is more robust. It uses only a few particles to keep particle diversity successfully, which also improves computing efficiency. It is more precise in localization, avoiding falling into local optimum. It is robust in complex scenes such as serious occlusion, temporary shielding or obvious illumination change. Experiments show the robustness of ABMSPF, as well as its location precision and computing efficiency.

There are still many work needed to improve the tracking performance. For example, we use the color feature in this experiment, but other features such as contours or combination of several features can be used as substitute. The object module can be updated online, which will improve the system robustness. Selecting rational transition function for particles is an essential step which usually determines performance of the algorithm.

TABLE II: Kinect Localization Results

Frame Number	Similarity in Mean shift	X/mm	Y/mm	Z/mm
25	0.851	-267.127	163.5	663
26	0.856	-267.127	158.895	663
27	0.838	-266.778	155.909	665
28	0.838	-267.580	156.378	667
29	0.840	-272.275	157.081	670

References

[1] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid object using mean shift," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, South Carolina, USA, June 2000, pp. 142–149.

[2] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE transactions on pattern analysis and machine intelligence (PAMI)*, no. 5, pp. 603–618, May 2002.

[3] A. Yilmaz, "Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, Minnesota, USA, June 2007, pp. 1–6.

[4] X. Chen, C. Li, Y. Luo, and G. Li, "Adaptive bandwidth mean shift algorithm and object tracking," *Robot*, pp. 147–154, 2008.

[5] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *European Conference on Computer Vision (ECCV)*, 2002, pp. 661–675.

[6] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, 2010.

[7] B. Babenko, M. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Florida, USA, June 2009.

[8] M. Xue and H. Ling, "Robust visual tracking using l1 minimization," in *IEEE International Conference on Computer Vision (ICCV)*, Kyoto, Japan, 2009, pp. 1436–1443.

[9] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-n learning: Bootstrapping binary classifiers by structural constraints," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, San Francisco, CA, USA, June 2010, pp. 49–56.

[10] K. Zhang, L. Zhang, and M.-H. Yang, "Real-time compressive tracking," in *European Conference on Computer Vision (ECCV)*, Firenze, Italy, 2012.

[11] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Portland, USA, June 2013.

[12] C. Teuliere, L. Eck, and E. Marchand, "Chasing a moving target from a flying uav," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, September 2011, pp. 4929–4934.

[13] S. Izadi, D. Kim, and O. Hilliges, "Kinect fusion: Real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, Santa Barbara, CA, USA, October 2011, pp. 559–568.

[14] K. Khoshelham, "Accuracy analysis of kinect depth data," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, no. 5, pp. 133–138, 2011.