Towards Occlusion Handling in Visual Tracking-by-Detection Systems

Yao Yeboah, Zhuliang Yu, and Wei Wu

***Abstract*—Occlusion is a common problem encountered in various tracking applications. This paper addresses occlusion within the context of real-time tracking. Contributions of the paper are two-fold. Firstly, the paper studies the occlusion problem within the context of tracking-by-detection. Secondly, a 3D approach that allows for tracking-by-detection algorithms to be extended towards effective occlusion handling is put forward. The proposed approach achieves an efficient incorporation of depth features into the circulant tracking framework, thereby achieving rapid object detection, tracking and robustification. Finally, a patch-based modeling strategy for depth features, coupled with a robust occlusion estimator is proposed. The resulting scheme allows for the tracker to achieve occlusion detection, tracker recovery and a significant alleviation of the drift problem associated with tracking-by-detection state-of-the-art. Experimental results on benchmark sequences demonstrate the effectiveness and robustness of the proposed scheme. The superiority of the scheme is further established in comparison experiments with state-of-the-art.**

***Index Terms*—Circulant tracking, occlusion-handling, robust tracking, 3D tracking.**

I. Introduction

Various machine vision algorithms and systems apply visual tracking as a fundamental and crucial step, the result of which becomes the foundation for realizing high-level and complex systems. The applications of visual tracking are therefore not limited exclusively to machine vision as seen in biomedical imaging [1], [2], video security systems [3] and vision-based traffic monitoring [4], but further extend into more complex and highly integrated systems including but not limited to natural interaction [5], visual servoing [6] and anomaly detection [7]. The design of efficient and robust tracking algorithms continues to remain an ongoing effort due to the various challenges associated with the research area. Regardless of the application domain, visual tracking presents multiple challenges that need to be explicitly handled in order to realize efficient algorithm design. These challenges include illumination variations, occlusion, target deformation, pose variations, background clutter, in-plane rotation, but to mention a few.

In an attempt to tackle these challenges and realize robust visual object tracking, various approaches have resorted to domain-specific solutions that constrain the tracking problem to specific objects such as human limbs [5], hands [8] and vehicles [9]. While such approaches have achieved high efficiencies as well as remarkable tracking speeds, they fail to extend this success rate to arbitrary targets due to their immense dependence upon offline target modeling, a task that becomes practically unachievable when handling arbitrary objects. The need for a new tracking paradigm, one that is capable of capturing the various scene and object dynamics could not be exaggerated. Within this tracking-by-detection paradigm [10], contrary to offline model-based schemes, the tracker is initialized with the object location and state within the first frame of the video sequence. Upon successful initialization, the tracking task is defined as the successful estimation of the target location and state within subsequent frames until termination of the video sequence. This approach to tracking can therefore be regarded as an iterative detection operation, which is performed on a per-frame basis. This task quickly highlights the need for efficient online target modeling, online training and online-the-fly classifier update strategies. Generally speaking, tracking-by-detection algorithms have either approached the problem through target modeling [11], [12], motion modeling [12] or target appearance representation. Some approaches also attempt to achieve integration of multiple components into a single tracking framework [10]. In most tracking-by-detection schemes, object detection and tracking is treated as a binary discriminatory problem and this has motivated the application of various discriminative classification schemes towards the realization of solutions. Such detection schemes have proven to be simple but remarkably efficient and some significant contributions are presented in [11], [13]-[18]. It is crucial to point out that while all these approaches may collectively fall into tracking-by-detection, each of the approaches exclusively addresses the problem either as a modeling task, learning task or parameter and model update task. Some widely utilized learning schemes include Support Vector Machines [19], Structured Support Vector Machines [15], Booting [20] and Bayes Classification [21]. It has been established that local appearance modeling schemes have the capability of robustifying trackers against partial appearance changes [11], [22]. Such schemes are similar to the patch-based object modeling scheme presented here in the paper. However, the proposed scheme applies robust depth features in the realization of patch-based modeling, rather than adopting unstable and highly sensitive colour or intensity features as previous works have attempted. Dense sampling [15], [17]

Manuscript received November 5, 2016; revised January 23, 2017. This work was supported in part by the Western Transportation Construction Technology Project of the Ministry of Transport, P.R.China under Grant B1110210.

Yao Yeboah is with the Department of Electrical and Computer Engineering, South China University of Technology, Tianhe, Guangzhou, China (e-mail: yeboahjunior@yahoo.com).

Zhuliang Yu and Wei Wu are with the School of Automation Technology, South China University of Technology, Tianhe, Guangzhou, China (e-mail: zlyu@scut.edu.cn, auweiwu@scut.edu.cn).

doi: 10.18178/ijfcc.2017.6.1.480

which has offered a radical approach to target detection, has been proven to outperform sparse sampling which fails to handle rapid target motion and background clutter [23]. This distinguishing condition is attributed to the capability of dense sampling schemes to adaptively scale the search domain on-demand to keep up with various motion characteristics.

Most real-time systems require tracking to keep up with real-time application demands. In satisfying the real-time demands of trackers, a majority of efforts have focused on applying 2-dimensional machine vision schemes that harness multiple features and apply learning strategies that are only capable of conceptualizing and interpreting 2-dimensional information. While such systems have indeed been light-weight and sufficient in achieving real-time results, a majority of them are left prone to noise corruption and fail to explicitly handle occlusion; a common drawback in most state-of-the-art trackers. A typical example is seen in the Circulant Structure Kernel Tracker (CSK) [18], which is capable of achieving remarkably fast and efficient tracking results without trading off kernel simplicity; a crucial requirement in maintaining tracking speed for real-time systems. Despite its speed and efficiency, the CSK tracker is left prone to corruption resulting from occlusion and this severely hampers its performance in most applications, causing it to drift in most partial occlusion situations, without the possibility of recovery. In most real-time applications, especially in robotics applications, scene dynamics could be unconstrained and there is the need for trackers that are capable of handling drift, regardless of sequence length. Thus the motivation of the work in this paper is established. Depth features have been extensively exploited in machine vision as a means of robustification while guaranteeing sustained efficiencies in the presence of noise. To the best of our knowledge however, efforts to harness this depth robustness within the tracking paradigm are very limited and lacking. 3D and depth imaging devices have become more abundant and affordable in recent years and the potential of efficiently incorporating depth into the tracking framework is significant and merits further research.

In addressing this problem, this paper proposes an occlusion robustification strategy for state-of-the-art via efficient depth incorporation and occlusion estimation. In the proposed scheme, depth features are efficiently incorporated into the CSK tracker without significantly trading off kernel simplicity and tracking speed. This incorporation is achieved through feature-warping and feedback strategies between depth and RGB spaces coupled with an intermediary stage between object detection and model update; a scheme that addresses the naïve parameter update in the CSK tracker. Furthermore, in guaranteeing robustness in this new RGBD tracker, local patch-based object modeling is proposed within the depth space. This patch-based local modeling is proven to be sufficient in handling partial occlusion as well as object deformation. Object deformation and extensive work on the RGBD feedback strategy as well as the proposed solution to naïve parameter updates in the CSK however fall outside the scope of this paper. Brief introductions will however be provided for clearer proofs and demonstrations. This paper focuses on the efficient depth-based object modeling problem within the proposed RGBD circulant tracker, by treating the tracking task as an object modeling problem. Experimental results attained on various challenging scenarios demonstrate the robustness of the proposed scheme to occlusion, leading to drift-alleviation while significantly maintaining the tracking speed characteristic of circulant structure kernels. The remainder of the paper is thus organized. Related work is presented in the Section II. The proposed RGBD tracking scheme is theoretically presented and discussed comprehensively in the Section III. Section IV covers experimental verification and benchmarking results with state-of-the-art. The paper concludes in Section V and future work is proposed.

II. Related Work

In order to achieve robust tracking performance in both tracking-by-detection and offline model-based trackers, considerable effort has been exerted and a number of tracking methods proposed. Earlier approaches have constrained the tracking problem by utilizing fixed object models [24], [25]. Such approaches address tracking by assuming that object appearance remains fixed across all frames constituting the video sequence and hence, the appearance model extracted from the first frame therefore remains viable for object detection in subsequent frames. While this assumption may hold true in some applications, the majority of scenarios come riddled with various scene and object dynamics and greatly challenge such approaches and render them inefficient for robust long-term tracking. Drawing from the drawbacks of these earlier approaches, relatively more robust tracking schemes have attempted to capture and model all possible variations of the target object offline, prior to tracker initialization. While such offline model-based schemes [10], [26] considerably alleviate the model rigidity problem associated with earlier schemes, their robustness depends heavily on pre-modeled scene and object dynamics. However, in most real life applications, such dynamics may be uncontrolled and so varying in nature that, attempting to model and store each variation becomes a practically infeasible task, the difficulty of which linearly correlates with increasing video sequence lengths. Drawing from this, the state-of-the-art have treated the tracking task as an online learning problem in which target appearance is modeled and updated online by exploiting target information from previous frames. Furthermore, these tracking-by-detection schemes [10], [27] have treated the object tracking problem as a detection problem stretching across all frames of the video sequence and have excelled greatly in achieving efficiency in situations where the target object remains unknown prior to initialization. Unfortunately however, such tracking schemes have suffered from drift; a gradual adaptation of the tracker to non-targets within the scene. While online trackers suffer from various challenges including object appearance variations, illumination changes, deformations and pose variations [16], [23], most of these challenges inadvertently result in tracker drift, which remains a core problem in tracking-by-detection systems.

In order to alleviate drift and robustify trackers, various schemes have been proposed. In [28], a method for ensuring tracker stability and constraining its motion using initial object appearance is proposed. In a similar manner, the work in [29] attempts drift mitigation by treating all incoming training samples as unlabeled data in a semi-supervised learning manner. While such schemes may suffice in handling moderate variations within the target scene, large variations may still destabilize the tracker. By harnessing context information abundant within the target scene, the work in [16] explicitly defines and relies upon so-called distractors and supporters in ensuring that the tracker does not gradually adapt to other objects in place of the target. Here also, significant robustification against drift is achieved but the tracker still fails to maintain integrity when rapid appearance changes and articulated poses are encountered. Through implementation with multiple instance boosting and co-training respectively, the works in [30] and [31] offer some solution to the drift problem. While these methods attain some degree of drift alleviation, occasional drifts still occur and high tracking speeds are sometimes traded-off; a limitation to feasibility in real-time tracking applications. A model-adaptation strategy in presented in [32]. The proposed strategy is driven by a feature-matching scheme that is robust to tracker drift. Drift is further addressed in [33] with the proposal of another method capable of automatically estimating the degree of local (dis)order within a target object. All these approaches fall within 2-dimensional tracking, leading to a lack of sufficient feature-level robustness, which may propagate throughout the tracking system.

Within the context of object tracking, some previous attempts have been made in attempting to achieve robustness through 3-dimensional frameworks. A 3D-based principal axis stereo tracker is proposed in [34]. A multi-target approach is proposed in [35] in which depth features are relied upon in verifying candidate objects selected via object detection. Some domain-specific 3D trackers are seen in [36], [37]. While such approaches succeed in harnessing depth robustness towards realizing some degree of drift alleviation, they perform sub-optimally in outdoor scenes where complicated illumination variations may occur. Furthermore, their domain specificity to human targets limits their adaptation and application potential to arbitrary target tracking.

III. Overview of Proposed Algorithm

This section presents the proposed scheme in detail. Drawing from the proven robustness of depth features to occlusion, the proposed algorithm adopts a 3D approach to tracking which effectively combines and exploits both RGB and depth features within a single tracking framework. Within this framework, the CSK tracker is adopted as a core tracker due to its lightweight and simple kernel structure, allowing it to sustain tracking speeds well beyond most state-of-the-art trackers. The simple kernel structure and interface associated with the circulant tracker allows for straight-forward extensions to be realized with minimal trade-offs. This motivates its selection for demonstrating the proposed scheme. Upon tracker initialization with the target coordinates within the first frame of a video sequence, a simple handshake process achieves a synchronization of target coordinates in both RGB and depth space. As experimental results will show, this hand-shake offers significant advantages in tracking speed and simplicity, compared with some earlier approaches that attempt to execute parallel trackers within both RGB and depth space [38]. Upon successful initialization in the first frame, circulant tracking begins and candidate training samples are extracted towards classifier training. Simultaneously, a local patch-based appearance modeling of the target object is conducted within depth space. Due to a core limitation in Time-of-Flight (ToF) sensing however, depth pre-processing and a patching up of *depth holes* is required before any high level operations can successfully exploit features within the depth stream acquired via the Kinect's ToF sensors. Here in this paper, depth-based Gaussian Mixture Modeling (GMM) [39] is adopted in the realization of depth sequence optimization. Following depth optimization, patch-based modeling is then applied to the optimized depth sequence. At this stage, a robust occlusion estimator which exploits the patch-based model obtained form depth space, conducts occlusion detection and adaptive model updates that aim to effectively capture the true feature representation of the target in both non-occlusion and occlusion states. By means of this adaptive-modeling, occlusion recovery becomes feasible in subsequent frames of the video sequence. The robust occlusion estimator precedes classifier training in order to alleviate the problem of naïve classifier learning in occlusion scenarios, thereby ensuring tracker integrity before, during and after occlusions. This 3D occlusion-robust tracking scheme iterates over all frames until termination of the video sequence. The proposed algorithm is graphically depicted in the Fig. 1.

A. Depth-Based GMM Optimization

The proposed occlusion-robust tracker presented in the paper exploits depth features in the realization of a 3D tracking framework, a strategy that allows the tracker to harness the robustness of depth features.

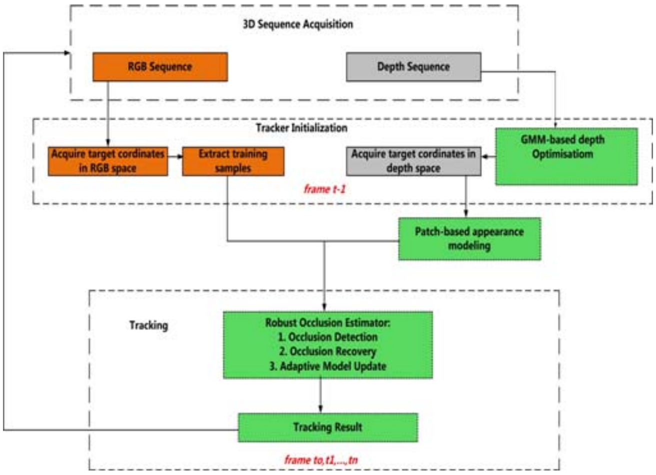


Fig. 1. Functional components of the proposed tracker.

While 3D sensing is achievable through various techniques, the ToF technique [40] offers significant advantages within the tracking paradigm due to its compact design and ability to realize 3-dimensional scene reconstruction without the need for baselines and multi-camera systems. Despite the obvious advantages associated with this 3D sensing scheme, depth holes or otherwise referred to as depth shadows, remain a core drawback [41]. The Fig. 2 graphically illustrates this problem which leads to ambiguities within the depth spectrum, and without effective preprocessing of this noisy depth sequence, tracking efficiency and classifier stability are put at risk. Within real-time applications, multiple foreground objects may exist within the scene and this produces multimodal characteristics that need to be effectively handled within the acquired depth sequence prior to tracking.

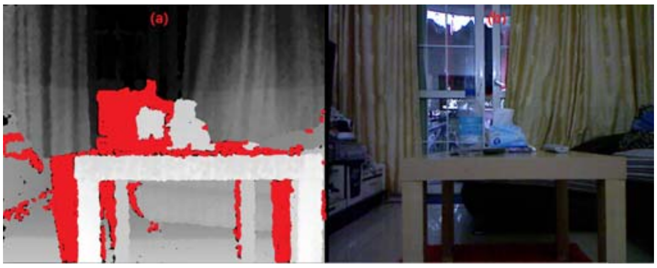


Fig. 2. Depth shadows remain a core drawback in ToF sensing techniques. Depth shadows are represented by red pixels in (a) while (b) illustrates the corresponding RGB image.

In the proposed tracking scheme, the Gaussian Mixture Model (GMM) is subverted in achieving pre-processing and optimization of the raw depth sequence acquired via the ToF sensor. The depth-based GMM considers the distribution of each pixel within the depth sequence towards the establishment of a reliable mixture model which defines the pixel variations over multiple frames. This modeling scheme is capable of distinguishing between foreground and background objects by relying on the variance and weights as distinguishing features. The algorithm operates as follows; Firstly, the Gaussian models are explicitly defined upon tracker initialization and the Yilmaz distance is computed. Secondly, each pixel within the sequence is independently processed and compared with the most current best model. If a match is established, the pixel is then classified as belonging to the model and model update is performed. However, if a pixel is determined to diverge from the model, a new GMM model is defined for this divergent pixel and parameters are reinitialized, causing the old unreliable model to become discarded. Finally, the best background model is selected towards realization of foreground segmentation.

Assuming that the features of each pixel within the depth sequence are expressed in a *K* Gaussian model, then after a new frame is acquired, the GMM is updated and the current pixel under consideration is matched with the GMM. If a match is attained, then the pixel is treated as the background, otherwise, the pixel is considered as the foreground. By representing the pixel feature as X_t, the probability distribution function of the Gaussian function could therefore be expressed by (1).

$$p(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{1}$$

where $\omega(i,t)$ represents an estimation of the weight of the ith Gaussian function at a time t. $\mu_{i,t}$ and $\Sigma_{i,t}$ represent the mean and variance of the i th Gaussian within the mixture at a time t where η is the Gaussian probability density function. This density function can in turn be expressed by (2) below.

$$\eta(X_t, \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|\Sigma_{i,t}\right|^{\frac{1}{2}}} \times e^{-\frac{1}{2}(Xt-\mu_{i,t})^T \Sigma_{i,t}^{-1}(X_t-\mu_{i,t})} \tag{2}$$

At the depth-based GMM optimization stage of the algorithm, the first frame of the depth sequence is applied in computing the frame difference iteratively. This yields a background update strategy that allows for the background depth values to be used in filling up holes present within incoming frames. This depth preprocessing and optimization scheme proves sufficient in addressing ambiguities and boosting the features within the depth sequence, a crucial step in ensuring tracker stability and integrity.

B. Adaptive Circulant Structure Kernel Tracking

The Circulant Tracker which was originally proposed in [18] is a renowned light-weight and highly efficient tracker with the capability of attaining the highest tracking speeds amongst the state-of-the-art [23]. This remarkable performance is attributed to the tracker's ability to exploit circulance in the structure attained when all periodic local patches are selected for classifier training. Despite its groundbreaking performance, the CSK tracker suffers from two core drawbacks. The first drawback is associated with the trackers naïve classifier training and parameter update strategy which predisposes the classifier to drift as depicted in the Fig. 3, while the second drawback is associated with the classifiers inability to detect and therefore effectively handle occlusions along with other significant object changes. The occlusion drawback is addressed with the patch-based modeling and occlusion estimator presented in the following section. However, the naïve classifier training and parameter update drawback is addressed here in this section.

In its simplest form, the CSK tracker applies a basic target model $\hat{x}$ with a classifier coefficient *A*. A linear interpolation strategy is then applied in updating the classifier parameters as follows in (3).

$$\alpha^{p} = (1-\gamma)\alpha^{p-1} + \gamma\alpha \tag{3}$$

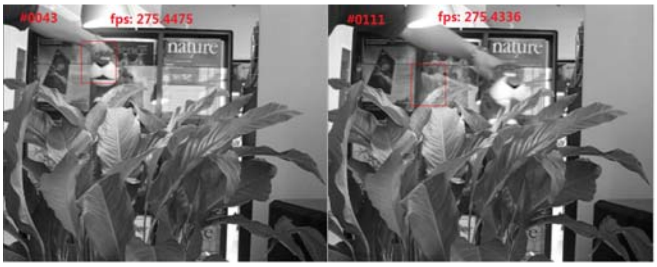


Fig. 3. The CSK tracker attains remarkable speeds but highly prone to *drift*.

where γ represents the learning rate parameter and *p* is the index of the current frame. The problem associated with such a basic learning scheme is two-fold. Firstly, the linear interpolation scheme fails to capture previous appearances of the object and hence, vital object features become discarded over time [42]. Additionally, this straight-forward interpolation constrains the learning rate to a fixed value, thereby trading off the flexibility required in allowing the tracker to adapt to the scene dynamics encountered in real-world scenarios. The parameter update strategy is therefore redefined as follows. A fixed weight of $\alpha_k \geq 0$ is set for each k th frame and this produces a cost function as illustrated in (4).

$$\vartheta = \sum_{k=1}^{p} \alpha_k (\Sigma_{m,n} \left| \left\langle \phi(x^k_{m,n}), w^k \right\rangle - y^k(m,n) \right|^2 + \lambda \left\langle w^k, w^k \right\rangle) \quad (4)$$

The learning rate parameter offers a means of adjusting the weights of the frames. This in turn yields an extended classifier parameter update strategy, which can be expressed by (5).

$$\alpha^p_N = (1-\gamma)\alpha^{p-1}_N + \gamma Y^p U^p_x \quad (5a)$$

$$\alpha^p_D = (1-\gamma)\alpha^{p-1}_D + \gamma U^p_x U^p_x (U^p_x + \lambda) \quad (5b)$$

In this manner, the target appearance model can be expressed by (6).

$$\hat{x}^p = (1-\gamma)\hat{x}^{p-1} + \gamma x^p \quad (6)$$

This extended update scheme offers a means of exploiting target appearance over all frames without explicitly storing all past models. Furthermore, this strategy overcomes the rigidity of the original CSK tracker by allowing the tracker to harness information within the problem domain towards effective and adaptive classifier learning.

C. Patch-Based Modeling and Occlusion Estimation

Appearance modeling has been considered as not only a core component of tracker design but also a crucial mechanism with which tracker robustification can be achieved. This section proposes a highly adaptive appearance modeling strategy realized within depth space. In realizing this goal, multiple non-overlapping local patches are segmented from each of the depth frames that constitute the target video sequence. In order to avoid redundancies, we assume that the bounding box captures the true location of the target within each frame and therefore designates the region of interest from which local patches are extracted. The problem is therefore formulated as follows; Having obtained target location designated by bounding box, *BB*, of width W and height H, divide the bounding box *BB* such that $BB' = (\alpha \times W) \times (\beta \times H)$ where α and β are the adjustable scale factors. The Fig. 4 below graphically depicts this strategy.

While existing 3D tracking schemes have been proposed to harness depth features towards robust tracking, feature extraction in such schemes has been carried out in a holistic manner which only exploits the overall appearance representation of the target within a single depth frame without explicitly capturing and modeling local depth feature variations of the target. This strategy renders the depth model of the target highly unreliable in partial occlusion situations, which may only manifest within sub-regions of the depth frame. We therefore propose to explicitly extract depth features from all local patches that constitute the bounding box. While various depth feature extraction schemes exist, including but not limited to Local Ternary Patterns (LPT) and Histogram of Oriented Vectors (HONV) [43], we adopt depth histograms as representative features of each local patch. This simple feature representation allows for the true nature and performance of the proposed tracking scheme to be observed in experimental results.

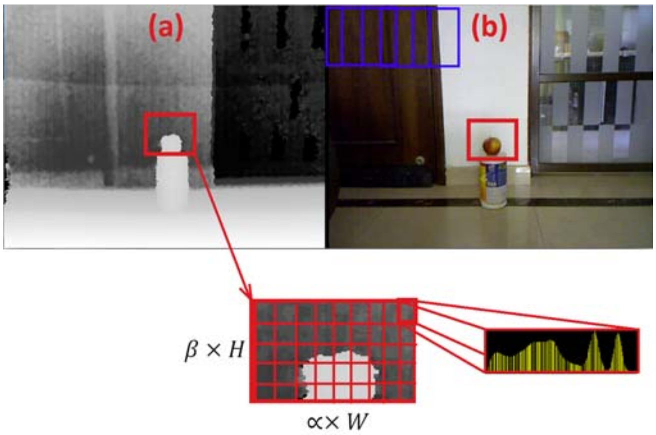


Fig. 4. A graphical illustration of (a) local patch-based target modeling strategy in depth space coupled with (b) Dense Sampling in RGB space (represented by blue overlapping sub-windows).

This patch-based target representation operates in unison with the circulant tracker by offering a feedback mechanism with which the CSK intuitively detects and efficiently handles occlusions. As depicted in the Fig. 4, while patch-based target modeling is performed in depth space towards occlusion estimation, dense sampling is carried out in RGB space towards classifier training in a circulant manner. Dense sampling allows for the efficient exploitation of redundancy within training data. This concept is formulated in the (7). Given a single one-dimensional image, x, which can be expressed as an $n \times 1$ vector, the training samples acquired through dense sampling can be expressed as:

$$x_i = p^i x, \forall_i = 0,...,n-1 \quad (7)$$

where p represents the permutation matrix which cyclically shifts the vectors by a single element. In an intuitive manner, all samples become possible translated versions of x in an ideal case. This yields the circulance required for features to be evaluated rapidly within all sub-windows via the Fast Fourier Transform (FFT). The interested reader is referred to [18] for detailed theoretical proofs and further reading on circulant tracking.

The final stage in the proposed tracking framework consists of the robust occlusion detection, estimation and recovery. The robust occlusion detection and recovery exploits the robust depth features and the local patch-based depth modeling and operates as follows. Given a specific patch in depth space, if its depth variation between consecutive depth frames within the sequence exceeds a preset threshold, the local patch is considered as unstable. Otherwise, we treat the patch as stable. An unstable patch is treated as a local region within the target region, which may be undergoing occlusion or rapid appearance changes, and therefore the goal is to adaptively update the overall target model by discarding all unstable patches while retaining stable ones. In this way, the classifier avoids naïve parameter updates and refrains from learning corrupted models which will ultimately lead to tracker drift. We therefore seek to represent the target with all stable patches, N_{sp}. In determining patch stability, depth histogram intersection is relied upon in computing the degree of variation between two consecutive patches. A single element $M(x,y)$ within the likelihood map denoted as $M_i(\cdot,\cdot)$, is derived by computing the depth histogram intersection distance between a candidate patch centered at (x,y) and the ith patch. The various elements are finally fused together to generate a likelihood map. The robust occlusion estimator is then formulated by (8) as:

$$(x^*,y^*)=\arg\max_{(x,y)} S(x,y) \tag{8}$$

In the (8), $S(x,y)$ represents the correlation coefficient obtained within the sorted set of $M_i(x,y) \mid i=1,2,...,N_{sp}$. Building upon the robust occlusion estimation strategy, occlusion is recovered as follows. Upon detecting occlusion, the target model remains fixed at local regions with unstable patches. This allows for model integrity to be guaranteed during occlusion. Then, for all subsequent frames, repeat this adaptive model update until the minimum feature distance drops below the preset threshold for all unstable local patches.

IV. Experimental Evaluation and Discussion

This section presents experimental evaluation results of the proposed tracking scheme as well as comparison results with state-of-the-art. While the proposed tracking approach is aimed at real-time applications such as robotics, real-time computer-based experimental are adopted towards evaluation since the scope of this paper is constrained to the problem domain rather than the computational and implementation platform. Furthermore, the deployment platform upon which the algorithm is realized should have little to no impact on the overall performance and robustness of the scheme.

A. Experimental Setup

In both actual tests and benchmarking experiments carried out to evaluate the performance of the proposed tracking scheme with state-of-the-art, two adopted metrics are relied upon. The first metric applied is the success rate (SR), which is formulated in (9) as:

$$SR=\frac{area(BBR_G \cap BBR_T)}{area(BBR_G \cup BBR_T)} \tag{9}$$

In (9), BBR_G denotes the bounding box region established by the ground truth while BBR_T represents the bounding box region obtained by the tracker. Similar to the scheme applied in [44], the obtained *SR* is compared to a present threshold and if the *SR* exceeds this threshold, the track is considered successful and if not, registered as a failed track. The second metric adopted in the benchmarking experiments is the Center Location Error (CLE), which defines the Euclidean distance between the center of the obtained bounding box and that of the ground truth.

In an ideal case, benchmarking of a tracking approach, which is realized though tracking-by-detection, would be required to compare performance with state-of-the-art including TLD [17], CSK [18], STRUCK [15] and MIL [14]. However, an obvious challenge arises which lies in the fact that most state-of-the-art trackers are not explicitly designed to operate on 3-dimensional feature spaces and hence a direct comparison of the proposed occlusion-robust tracker with such state-of-the-art would be an ill-formulated problem. The Mean Shift tracker [38], and the Sparse Flow Tracker [45], which have been well established against other state-of-the-art are selected for benchmarking. Furthermore, these trackers have the capability of being extended to operate on RGBD sequences, which allows for fairness to be guaranteed in benchmarking. Furthermore, the original CSK algorithm is included in benchmarking to operate only on the RGB sequences that make up the benchmarking sequences. The details of the platform for computer-based experiments is as follows:

1) Computing platform: 2.8 GHz Intel Core i7
2) Memory: 8 GB RAM @1067 MHz)
3) Data Acquisition Scheme: Kinect Sensor@ 30FPS

In facilitating the actual and benchmark experiments, RGBD video sequences are captured and recorded with varying forms and degrees of complexity. While the proposed tracking scheme is explicitly designed to detect and effectively handle partial occlusions within a circulant tracking framework, the benchmarking sequences are recorded with varying forms of challenge in order to establish the impact of these various real-time challenges on the overall tracker performance. Table I presents details of the recorded sequences.

TABLE I: Benchmark Sequences and Their Associated Challenges

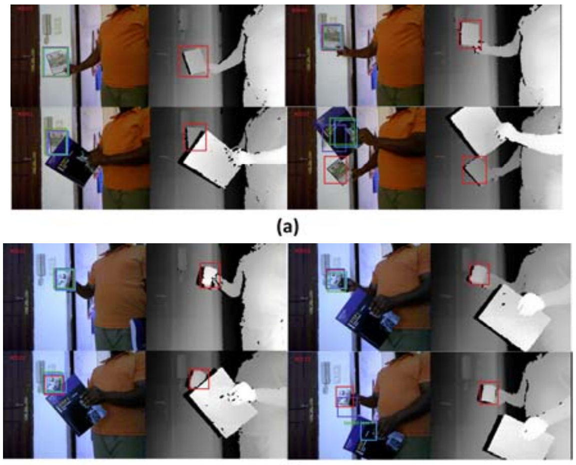
Sequence	Associated Challenges
1. Magazine	Partial Occlusion, Full Occlusion, Fast Motion
2. Mug	Partial Occlusion, Random Motion, Illumination Changes
3. Notebook	In-plane-rotation, Partial Occlusion
4. Palm	Out-of-view, Random Motion, Partial Occlusion

B. Experimental Results

Fairness in comparison experiments is guaranteed by ensuring that all trackers are spatio-temporarily initialized.

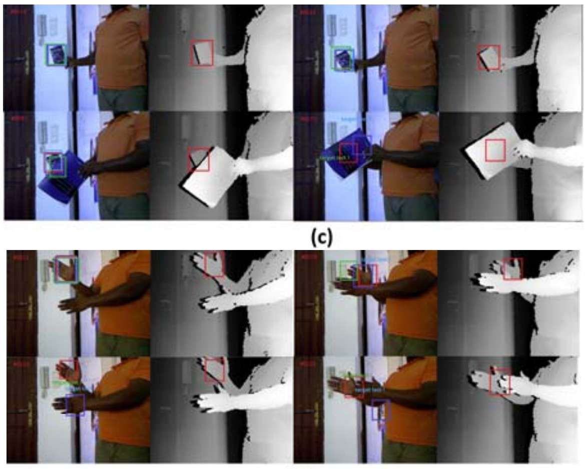
From the results depicted in the Fig. 5, all trackers are able to maintain some degree of stability in initial frames of each sequence. Intuitively, performance begins to vary with increase in sequence length. The CSK tracker, which operates only on RGB sequences exhibits the fastest performance which could be attributed to its capability to exploit circulance towards fast and efficient feature extraction and hence rapid iterative detection across frames. As the experimental results show, this simple approach to tracking suffers immensely when various complexities are introduced into the sequence. While partial occlusion leads to some degree of tracker drift, sustained occlusion of the target causes the tracker to become completely adapted to the occluding object and the impact of this instability only becomes compounded with an increase in sequence length. This drawback is attributed to the inability of the tracker to detect instances of occlusion, which in turn leads to a gradual degradation of the tracker in both partial and full occlusion situations. Over all sequences, this remains the area in which the proposed tracking scheme outperforms CSK. By decomposing the tracking problem and introducing a local patch-based appearance modeling scheme within an RGBD tracking framework, the proposed tracking scheme is capable of efficiently detecting target changes which could be as a result of occlusion, target deformation, rotation but to mention a few. Through implementation with a robust occlusion estimator, instances of occlusion can be detected and recovered on a per frame basis enabling the tracker to maintain stability even over long video sequences. As depicted in the experimental results, while CSK maintains a higher tracking speed, the proposed scheme is capable of avoiding tracker drift even in scenarios where occlusion is coupled with rapid target motion. This highlights the feasibility of the proposed scheme in real-time applications. While the most stable tracking results are obtained by the proposed tracking scheme, the Mean Shift tracker attains the most unstable performance, being slightly out-performed by the Sparse Flow tracker.

The Sparse Flow tracker adopted in this paper adopts a pyramidal Kanade-Lucas Tomasi (KLT) core in realizing feature tracking within the bounding box region. Experimental results show that the tracker is highly prone to drift since it estimates target motion in relation to the previous frames.



(a)

(b)



(c)

(d)

Fig. 5. Visual tracking results achieved on (a) *Magazine* (b) *Mug* (c) *Notebook* (d) *Palm* sequences. Red: Proposed Tracking Scheme, Blue: CSK, Cyan: Mean Shift tracker, Green: Sparse Flow Tracker.

This estimation scheme means that over time, the tracker can be expected to have drifted from the true target position. While drift has not yet corrupted the tracker, tracking results are smoother than the Mean Shift tracker but not as fast as CSK and not as robust as the proposed scheme. The tracker is capable of handling minor occlusions but fails to recover after significant occlusions and target loss. The Mean Shift tracker which was adopted utilizes a fixed-size rectangular kernel which computes the likelihood of each pixel within the frame on a per-frame basis. This tracker attains a high tracking speed but tracking efficiency relies immensely upon the degree of visual distinctiveness between foreground and background pixels. Furthermore, tracker performance degrades in scenarios where foreground and background pixels may be visually distinctive but are not largely composed of one colour. The experimental results presented in the Table II demonstrate the high robustness of the proposed tracking scheme.

In terms of stability, the proposed tracking scheme outperforms state-of-the-art and achieves the highest precision and success rate over all sequences. The average error rate of the tracker is significantly smaller than state-of-the-art although the CSK outperforms in the *notebook* sequence.

TABLE II: PROPOSED TRACKING SCHEME VS. STATE-OF-THE-ART. SUCCESS RATE (SR) AND CENTER LOCATION ERROR (SLE)

Sequence/Frames	CSK	Mean Shift	Sparse Flow	Proposed
	SR/CLE	SR/CLE	SR/CLE	SR/CLE
Magazine/300	17.2/33.7	12.3/38.5	14.8/37.3	28.0/32.1
Mug/450	19.7/25.9	13.8/36.1	15.5/35.0	33.2/23.8
Notebook/400	26.5/18.3	24.2/23.5	25.0/21.7	31.8/19.3
Palm/400	28.2/12.2	22.6/27.6	24.9/27.0	37.2/9.5

The stable performance of the proposed tracker is attributed to the fact that apart from being robust to illumination variations due to its RGBD tracking approach, the scheme is further able to detect occlusion instances through implementation with patch-based local appearance modeling in depth space. This allows for selective and adaptive classifier training to be achieved and hence classifier integrity guaranteed. Furthermore, by holding off on model updates in regions of model instability, occlusion recovery is feasible when the target reemerges from occlusion as seen in *mug* #0115 and *magazine* #0107.

V. Conclusion

Occlusion is a predominant problem encountered by tracking-by-detection algorithms and systems. The destabilizing impact of occlusion on tracking is not limited to classical algorithms but extends to the state-of-the-art as well. In tracking-by-detection algorithms where tracking is treated as an iterative detection task stretching across all frames of a video sequence, the impacts of occlusion may include, but are not limited to tracker drift and complete tracker loss. This paper presents a tracker robustification scheme, which effectively incorporates local patch-based target appearance modeling and robust occlusion estimation and recovery into an RGBD tracking framework. The local patch-based target modeling allows for adaptive object modeling to be realized within depth space, a scheme that allows for model degradation to be mitigated in instances of occlusion. By harnessing this adaptive model update, a robust occlusion estimator is realized with the capability to detect occlusion on a per-frame basis and hence, allow for the tracker to be recovered when the object emerges from occlusion.

The proposed tracking scheme is implemented using CSK as a core tracker due to its lightweight and simple kernel interface. The robustness of the scheme to partial occlusions as well as its capability to recover in instances of full occlusion is verified through computer-based experiments realized with the Kinect sensor. Comparison experiments with state-of-the-art demonstrate the superior performance of the proposed scheme in terms of tracker robustness. Furthermore, due to the efficient incorporation of GMM-optimized depth features with RGB features into a single RGBD tracking framework, the tracker performance is not impeded upon by background clutter and illumination variations.

Future work will study the impacts of target deformation and in-plane rotation on the proposed tracking scheme, as well as extensions to robustify tracking against these challenges. Additionally, while computer-based experimental verification and benchmarking has been sufficiently carried out here in this paper, future work will cover experiments that simulate robot-based scenarios using the TurtleBot presented in Fig. 6 as an experimental platform.

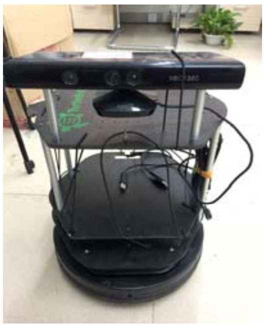


Fig. 6. The TurtleBot platform facilitates robot-based experimental evaluation of the proposed tracker.

Such robot-based experiments could include target tracking-and-following as well as target tracking-and-evasion scenarios.

Acknowledgment

The authors acknowledge the support of the RobotScut research lab of South China University of Technology.

References

[1] K. Cannons, “A review of visual tracking,” Technical Report CSE-2008-07, CA: York University, 2008.

[2] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Computing Surveys*, vol. 34, no. 4, pp. 1-45, Dec. 2006.

[3] P. Gorur and B. Amrutur, "Skip decision and reference frame selection for low-complexity H.264/AVC surveillance video coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 24, no. 7, pp. 1156-1169, July 2014.

[4] L. P. Perera, P. Oliveira, and C. G. Soares, "Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction," *IEEE Trans. on Intelligent Transportation Systems,* vol. 13, no. 3, pp. 1188-1200, Sept. 2012.

[5] C. Chang and W. H. Tsai, “Vision-based tracking and interpretation of human leg movement for virtual reality applications,” *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 11, no.1, pp. 9-24, Jan. 2001.

[6] C. Y. Tsai, C. C. Wong, C. J. Yu, C. C. Liu, and T. Y. Liu, "A hybrid switched reactive-based visual servo control of 5-DOF robot manipulators for pick-and-place tasks," *IEEE Systems Journal*, vol. 9, no. 1, pp. 119-130, March 2015.

[7] B. Ristic, "Detecting anomalies from a multitarget tracking output," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 50, no. 1, pp. 798-803, January 2014.

[8] M. A. A. Aziz, J. Niu, X. Zhao, and X. Li, "Efficient and robust learning for sustainable and reacquisition-enabled hand tracking," *IEEE Trans. on Cybernetics*, vol. 46, no. 4, pp. 945-958, April 2016.

[9] K. H. Lee, J. N. Hwang, and S. I. Chen, "Model-based vehicle localization based on 3-D constrained multiple-kernel tracking," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 25, no. 1, pp. 38-50, Jan. 2015.

[10] S. Avidan, “Support vector tracking,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, pp.1064–1072, 2004.

[11] X. Jia, H. Lu, and M. H. Yang, “Visual tracking via adaptive structural local sparse appearance model,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1822-1829, June 2012.

[12] J. Yang, B. Price, X. Shen, Z. Lin, and J, Yuan, “Fast appearance modeling for automatic primary video object segmentation,” *IEEE Trans. on Image Processing*, vol. 25, no. 2, pp. 503-515, Feb. 2016.

[13] A. D. Jepson, D. J. Fleet, and T. F. El-Maraghi, “Robust online appearance models for visual tracking,” *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI),* vol. 25, no. 10, pp. 1296-1311, Oct. 2003.

[14] B. Babenko, M. H. Yang, and S. Belongie, “Visual tracking with online multiple instance Learning,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 983-990, June 2009.

[15] S. Hare, A. Saffari, and P. H. S. Torr, “Struck: Structure output tracking with kernels,” in *Proc. IEEE International Conference on Computer Vision (ICCV)*, pp. 263-270, Nov. 2011.

[16] T. B. Dinh, N. Vo, and G. Midioni, “Context tracker: Exploiting supporters and distracters in unconstrained environments,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1177-1184, June 2011.

[17] Z. Kalal, J. Matas, and K. Mikolajczyk, “P-N learning: Bootsrapping binary classifiers by structural constraints,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 49-56, June 2010.

[18] F. Henriques, R. Caseiro, P. Martins, and J. Batista, “Exploiting the circulant structure of tracking-by-detection with kernels,” in *Proc. 12th European Conference on Computer Vision*, vol. 4, pp. 702-715, Oct. 2012.

[19] S. Avidan, “Support vector tracking,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol.1, pp. 184-191, 2001.

[20] S. Avidan, “Ensemble tracking,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 261-271, Feb. 2007.

[21] X. Chen and J. Wu, “Scalable compressive tracking based on motion,” in *Proc. IEEE International Conference on Robotics and Biometrics (ROBIO)*, Shenzhen, China, Dec. 12-14, 2013.

[22] W. Zhong, H. Lu, and M. H. Yang, “Robust object tracking via sparsity-based collaborative model,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, pp. 1838-1845, 2012.

[23] Y. Wu, J. Lim, and M. H. Yang, “Online object tracking: A benchmark,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2411-2418, June 2013.

[24] M. Isard and A. Blake, “Condensation-conditional density propagation for visual tracking,” *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5-28, 1998.

[25] D. Comanicu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 142-149, 2000.

[26] A. Adam, E. Rivlin, and I. Shimshoni, “Robust fragments based tracking using the integral histogram,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 798-805, June 2006.

[27] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631-1643, Oct. 2005.

[28] L. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 810-815, June 2004.

[29] H. Grabner, C. Leistner, and H. Bischof, “Semi-supervised on-line boosting for robust tracking,” in *Proc. European Conference on Computer Vision*, vol. 5302, pp. 234-247, 2008.

[30] B. Babenko, M. H. Yang, and S. Belongie, “Visual tracking with online multiple instance learning,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 983-990, June 2009.

[31] Q. Yu, T. Dinh, and G. Medoni, “Online tracking and reacquisition using co-trained generative and discriminative trackers,” in *Proc. European Conference on Computer Vision*, vol. 5303, pp. 678-691, 2008.

[32] L. Lu and G. Hager, “A non-parametric treatment for location/segmentation based visual tracking,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-8, June 2007.

[33] S. Oron, A.B. Hillel, D. Levi, and S. Avidan, “Locally orderless tracking,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1940-1947, 2012.

[34] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank, "Principal axis-based correspondence between multiple cameras for people tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI),* vol. 28, no. 4, pp. 663-671, April 2006.

[35] A. Ess, B. Leibe, K. Schindler, and L. V. Gool, "Robust multiperson tracking from a mobile platform," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 31, no. 10, pp. 1831-1846, Oct. 2009.

[36] S. Song and J. Xiao, “Tracking revisited using RGBD camera: Unified benchmark and baselines,” in *Proc. IEEE International Conference on Computer Vision*, pp. 233-240, 2013.

[37] M. Luber, L. Spinello, and K. O. Arras, "People tracking in RGB-D data with on-line boosted target models," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3844-3849, 2011.

[38] B. Amit and W. Michael, “RGB-D fusion with mean shift tracking,” *Dynamic 3D Imaging*, vol. 5742, Berlin: Springer, 2009, pp. 58-69.

[39] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 22, no. 8, pp. 747-757, Aug 2000.

[40] E. Tadmor *et al*., "Development of a ToF pixel with VOD shutter mechanism, high IR QE, four storages, and CDS," *IEEE Trans. on Electron Devices*, vol. 63, no. 7, pp. 2892-2899, July 2016.

[41] Z. Tauber, Z. N. Li, and M. S. Drew, "Review and preview: Disocclusion by inpainting for image-based rendering," *IEEE Trans. on Systems, Man, and Cybernetics*, Part C (Applications and Reviews), vol. 37, no. 4, pp. 527-540, July 2007.

[42] D. Martin, S. K. Fahad, W. Michael, and W. Joost, “Adaptive color attributes for real-time visual tracking*,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1090-1097, June 2010.

[43] L. Spinello and K. O. Arras, "People detection in RGB-D data," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3838-3843, 2011.

[44] J. Smisek, M. Jancosek, and T. Pajdla, "3D with kinect," *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 1154-1160, 2011.

[45] B. D. Lucas and T. Kanade, “An iterative image registration technique with and application to stereo vision,” *in Proc. 7th International Joint Conf. on Artificial Intelligence (IJCAI)*, pp. 674-679, 1981.

Yao Yeboah received the B.Eng. in the field of electronic information engineering from the Huazhong University of Science and Technology, Wuhan, China in 2011 and the M.Eng. from the South China University of Technology, Guangzhou, China in 2013. He is currently pursing his Ph.D. in the Department of Electrical and Computer Engineering, School of Automation Science and Engineering, South China University of Technology. His research interests include pattern recognition, intelligent systems and robotic vision.

Zhuliang Yu received the BSEE in 1995 and the MSEE in 1998, both in electronic engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China and the Ph.D. in 2006 from Nanyang Technological University, Singapore. He joined the Center for Signal Processing, Nanyang Technological University in 2000 as a research engineer, then as a Group Leader. In 2008, he joined the College of Automation Science and Engineering, South China University of Technology, China. He was promoted to be a full Professor in 2009. His research interests include signal processing, machine learning, computer vision and applications in biomedical engineering and robotics.

Wei Wu received the Ph.D. degree in the field of control theory and control engineering from the Huazhong University of Science and Technology, Wuhan, China in 2000. He is currently a professor in the School of Automation Science and Engineering of South China University of Technology, Guangzhou, China. His research interests include intelligent control systems, robotic control and engineering, pattern recognition and intelligent systems.