Constrained Confidence Matching for Planar Object Tracking

Tao Wang[1,3], Haibin Ling[2,3], Congyan Lang[1], Songhe Feng[1], Yi Jing[1] and Yidong Li[1]

***Abstract*— Tracking planar objects has a wide range of applications in robotics. Conventional template tracking algorithms, however, often fail to observe fast object motion or drift significantly after a period of time, due to drastic object appearance change. To address such challenges, we propose a novel constrained confidence matching algorithm for motion estimation and a robust Kalman filter for template updating. Integrated with an accurate occlusion detector, our approach achieves accurate motion estimation in presence of partial occlusion, by excluding occluded pixels from computation of motion parameters. Furthermore, the proposed Kalman filter employs a novel control-input model to handle the object appearance change, which brings our tracker high robustness against sudden illumination change and heavy motion blur. For evaluation, we compare the proposed tracker with several state-of-the-art planar object trackers on two public benchmark datasets. Experimental results show that our algorithm achieves robust tracking results against various environmental variations, and outperforms baseline algorithms remarkably on both datasets.**

I. INTRODUCTION

Tracking of planar objects, e.g. 2D markers, is often an important step in camera localization and scene registration, and has many applications in robotics [1], [2] and augmented reality [3], [4]. In this work, we address the problem in an accurate and robust manner, with arbitrary motion and no prior knowledge other than its position in the first video frame.

In the past few decades, a large mount of investigations were devoted to visual tracking problem. Popular approaches to planar object tracking can be roughly classified as keypoint-based approaches (e.g., [5], [6], [7]) or template-based ones (e.g., [8], [9], [10], [11], [12]). Template-based approaches directly use the appearance of the pixels without extracting features, and optimize a similarity measure between the template and the captured image, based on the Newton method or its variants, to determine the pose of the plane. Although the template tracking problem has been investigated for decades, it remains challenging due to various perturbations such as illumination change, motion blur and occlusion.

Conventional template-based trackers that hold fixed templates usually fail to observe the object motion in presence of drastic appearance changes that violate the *brightness constancy assumption* (BCA) [8]. A common way to improve the robustness is to update the template dynamically using some heuristic strategies [13], [14], [15] or online-learning methods [16], [17]. Another popular way is to replace the classical *sum of squared differences* (SSD) measure with new similarity measures which are more robust to brightness changes. Some samples include the *sum of conditional variance* (SCV) [18], [15], the local *normalized cross correlation* (NCC) [19], the *mutual information* (MI) [20], and the *gradient orientations* (GO) [12]. However, most of these approaches address the challenge from illumination change, and still suffer from other environmental perturbations, especially for partial occlusion and motion blur. Therefore, it demands more research attentions on how to develop more robust algorithms in presence of various perturbations.

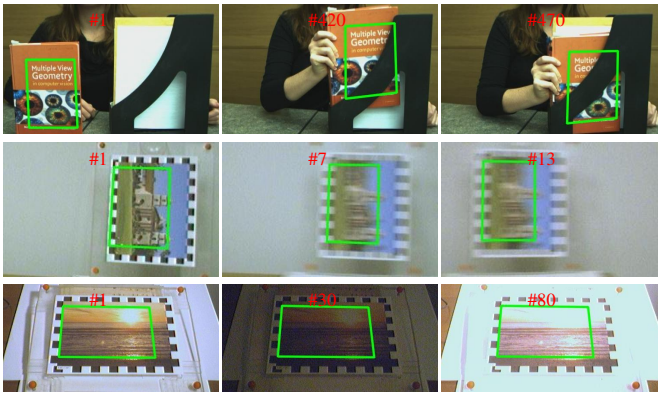

Fig. 1. Example results of the proposed CCM algorithm in presence of drastic appearance changes including partial occlusion (the first row), motion blur (the second row) and lighting change (the last row). We only show zoomed-in regions around the target for better illustration. Green boxes indicate tracking results of the algorithms, and *no box* indicates missing of target. The frame index is shown on top of each image, and the first frame shows the initialization. Same styles are used in Figures 2 to 5.

In this paper, we propose a *constrained confidence matching* (CCM) algorithm to improve the robustness against various environmental perturbations including partial occlusion, illumination change and motion blur. Firstly, unlike conventional template-based trackers that treat each pixel in the template identically when computing the motion parameters, we assign pixels with different matching confidences according to the measurement noises. With the assistance from an accurate occlusion detector we proposed, our tracker gains high robustness against partial occlusion by excluding occluded pixels from computation of motion parameters. Furthermore, in order to improve the robustness against sudden illumination change and heavy motion blur,

[1]Tao Wang, Congyan Lang, Songhe Feng, Yi Jing and Yidong Li are with the School of Computer & Information Technology, Beijing Jiaotong University, Beijing 100044, China. {twang, cylang, shfeng, yjing, ydli}@bjtu.edu.cn

[2]Haibin Ling is with Computer & Information Sciences Department, Temple University, Philadelphia, USA. hbling@temple.edu

[3]HiScene Information Technologies, Shanghai 201210, China.

we propose a robust Kalman filter that employs a novel control-input model to handle the appearance change of the object.

To thoroughly evaluate the proposed CCM algorithm, we test it on two public datasets: the *University of California, Santa Barbara* (UCSB) benchmark [21] and the *tracking manipulation tasks* (TMT) benchmark [22]. Experimental results show that, our tracker achieves robustness against various perturbation factors and outperforms significantly state-of-the-art trackers (including keypoint-based tracker [5], template-based tracker [23], [10], [24], [15], [12] and probabilistic tracker [16], [17]) in comparison. Fig. 1 illustrates some representative examples of the proposed CCM algorithm in presence of drastic environmental variations, including partial occlusion, motion blur and lighting change, which usually harm the tracking accuracy of conventional template tracking algorithms. The source code of the proposed approach will be made public at http://www.dabi.temple.edu/~hbling/code/ccm/index.html.

II. Related Work

Visual object tracking has been investigated for several decades as summarized in [25], [26]. Classical early work dates back to the LK algorithm [8] for 2D template tracking and the ICP algorithm [27] for 3D object tracking. In the following we sample some related ones that inspire our study on *template-based planar object tracking*.

Template tracking consists of moving, and possibly deforming, a template to minimize the difference between the template and the current captured image. The *sum of squared differences* (SSD) is commonly utilized as the similarity measure. Gradient descent is the most usual approach to template alignment, of which a variety of derivation algorithms such as difference decomposition [28] and linear regression [29] have also been proposed. Hager and Belhumeur [23] propose an *inverse compositional* (IC) algorithm to improve the efficiency of computing Hessian in each iteration by switching the role of the image and the template. This algorithm is further heavily discussed by Backer and Matthews [9]. Malis proposes an *efficient second-order minimization* (ESM) algorithm [10] that employs control laws based on the second-order Taylor series to achieve high convergence rates and avoid convergence problems of conventional Newton minimization method.

The above trackers use fixed templates, and thus usually fail to observe the object motion in presence of drastic appearance changes. One typical way to enhance the robustness of such object trackers is to use a pre-trained view-based eigenbasis representation [30], [31]. It is imperative for these approaches to collect a large set of training images covering the range of possible appearance variations to construct the eigenbasis as the representation. In the case of less prior knowledge of the target, a more proper way is to update the template dynamically. A naive strategy is to directly replace the template every frame with the tracking result. In [13], the authors propose to retain the first template as well as maintaining a current estimate of the template. The template is first updated as in the naive algorithm and then aligned with the first template to give the final update. For estimating the intensity of the template in a more accurate manner, Kalman filtering is adopted to track the template intensity under a Gaussian system [32]. This work is followed by Nguyen et al. [14], [24] who combine occlusion detection with Kalman filtering to reject occlusion from template updating. Lacking of the ability to model environmental inputs, this approach tends to reject them as outliers from template updating, and thus suffers from drastic environmental variations including motion blur and lighting change. In [33], [17], an incremental PCA model is adopted for online-learning of the appearance model of the object.

In addition to adapting the template for appearance changes, another popular way is to apply robust similarity measures to replace SSD. *Mutual information* (MI) [20] and *cross cumulative residual entropy* (CCRE) [34] originally applied in medical image registration are successfully introduced into visual tracking to improve the robustness against illumination changes. Richa et al. propose a new similarity measure, named *sum of conditional variance* (SCV) [18], aiming to cope with non-linear illumination changes. This approach maintains a joint intensity distribution of the template, and updates it in every new frame using the most recent observation. This work is further extended in [15]on decreasing its sensitivity to local illumination changes. More recently, the *gradient orientations* (GO) feature [12] is introduced into template tracking to handle complex illumination changes. By using GO features, the authors generalize the original ESM algorithm to a GO-ESM algorithm for multi-dimensional features. However, most of these algorithms address only the challenge from illumination change, but still suffer from occlusion and motion blur.

Alternative to the above mentioned approaches based on energy minimization, learning-based approaches appear to be robust to environmental perturbations. Sampled recent studies include [16] that decomposes the long-term tracking task into tracking, learning, and detection; [35] that adopts random forests to learn the relation between the motion parameters and the changes on the image intensities; and [17] that formulates the template-based visual tracking problem as a particle filtering problem on the matrix Lie group. A common shortage of this category of approaches lies in that they usually fail to acquire accurate motion estimation in presence of extreme pose changes.

The proposed CCM algorithm is inspired by the above algorithms in the enhancing of similarity measure and the updating of template to assist tracking, but differs in the way of capturing environmental inputs and rejecting occlusion from matching and updating. CCM aims to provide robust and accurate tracking for planar objects against various environmental variations, and the excellent experimental results clearly validate its advantage.

III. Template-based pose tracking

Suppose we are given a video sequence of images $I_t(\mathbf{x})$ where $\mathbf{x} = (x, y)^\top$ are the pixel coordinates and $t = 0, 1, 2, \ldots$

is the time moment. Pose tracking aims to acquire the pose Φ_t of the object of interest in each frame t, or to report the object missing when it is invisible. The relative motion between the object and the camera induces changes in the position of the object in the image. We assume that these transformations can be modeled by a geometric warping $\varphi(\mathbf{x};\mathbf{p}_t):\mathbb{R}^d\rightarrow\mathbb{R}^d$, where $\mathbf{p}_t$ denotes the parameter vector of the transformation specified for Φ_t. For 2-D transformation, φ is usually defined as an affine transformation. Considering *eight degrees of freedom* (8DOF) pose tracking of planar objects in this paper, we define φ as a perspective transformation.

In template-based tracking, a region that contains the object of interest is extracted and becomes the template T, where $T(\mathbf{x})$ denotes the intensity of pixel $\mathbf{x}$ in the template. The initial template T is usually given in advance or is picked from the first frame. The goal of template tracking is to find the best match to the template in every subsequent frame. Using SSD as the dissimilarity measure, the template tracking problem can be formulated as finding optima $\mathbf{p}_t$ to minimize the dissimilarity function

$$\min_{\mathbf{p}_t}\mathscr{E}_1(\mathbf{p}_t;I_t)=\sum_{\mathbf{x}\in\wedge_T}[I_t(\varphi(\mathbf{x};\mathbf{p}_t))-T(\mathbf{x})]^2, \tag{1}$$

where $\wedge_T$ is the support of T.

Some conventional approaches simply fix the template T across all video frames, while recent approaches tend to adapt it dynamically for appearance changes.

IV. Constrained confidence matching against occlusion

A. Constrained confidence matching

Conventional template-based algorithms usually treat each pixel in the template identically when computing the motion parameters, and are thus sensitive to some extrinsic noises, especially for occlusion.

To address this issue, we propose a *confidence matching* strategy that assign pixels with different matching confidences in computing the motion parameters. Intuitively, we assign low confidences to the pixels interfered by noises. Taking matching confidences into account, the matching dissimilarity defined in Eq. (1) is extended as

$$\begin{aligned}\min_{\mathbf{p}_t}\mathscr{E}_2(\mathbf{p}_t;I_t)&=\sum_{\mathbf{x}\in\wedge_T}C(\mathbf{x})[I_t(\varphi(\mathbf{x};\mathbf{p}_t))-T(\mathbf{x})]^2.\\ &\text{s.t.}\quad \mathbf{b}\succcurlyeq\mathbf{p}_t-\mathbf{p}_{t-1}\succcurlyeq-\mathbf{b}.\end{aligned} \tag{2}$$

where C denotes the confidence map in which each entry $C(\mathbf{x})$ records the matching confidence of pixel $\mathbf{x}$, $\mathbf{b}\succcurlyeq\mathbf{0}$ denotes the tolerance of geometric changes, and $\succcurlyeq$ is the element-wise $\geq$. The constraints we added is to forbid any leap of motions between consecutive frames.

The confidence map C is initialized uniformly for each pixel $\mathbf{x}$, and is further updated per frame according to the difference between the previous observation and the template

$$C(\mathbf{x})=1-\frac{(I_{t-1}(\varphi(\mathbf{x};\mathbf{p}_{t-1}))-T(\mathbf{x}))^2}{\varepsilon^2}, \tag{3}$$

where ε denotes the maximum difference

$$\varepsilon=\max_{\mathbf{x}\in\wedge_T}|I_{t-1}(\varphi(\mathbf{x};\mathbf{p}_{t-1}))-T(\mathbf{x})|. \tag{4}$$

The tolerance of geometric changes $\mathbf{b}$ is learned adaptively according to the motion parameters of the previous k frames

$$\mathbf{b}=\frac{\rho}{k}\sum_{i=1}^{k}|\mathbf{p}_{t-i}-\mathbf{p}_{t-i-1}|, \tag{5}$$

while ρ is an amplification coefficient to the average motions of the previous k frames, $|\cdot|$ denotes the absolute value of a vector. We set $\rho=5$ and $k=20$ throughout our experiments.

B. Optimization

There are a large number of literatures dedicated to original matching problem (1), some samples include [9], [10], [17]. Here we adopt the ESM algorithm [10] and extend it to solve the constrained confidence matching problem (2).

Let us first consider the unconstrained confidence matching problem, e.g., dropping the constraints in problem (2). Denote $\mathbf{J}(\mathbf{p};I)$ the Jacobian of $\mathscr{E}_2$ evaluated at parameter $\mathbf{p}$ with image I, we have

$$\mathbf{J}(\mathbf{p};I)=\sum_{\mathbf{x}}\left[C(\mathbf{x})\nabla I\frac{\partial\varphi}{\partial\mathbf{p}}\right], \tag{6}$$

where $\nabla I=(\frac{\partial I}{\partial x},\frac{\partial I}{\partial y})$ is the gradient of image I evaluated at $\varphi(\mathbf{x};\mathbf{p})$, and the term $\frac{\partial\varphi}{\partial\mathbf{p}}$ is the Jacobian of the warp. For an incoming frame I_t, the motion parameter is initially estimated as $\mathbf{p}_t\leftarrow\mathbf{p}_{t-1}$. According to the *pseudo-inverse of the mean of the Jacobians* (PMJ) method proposed in [10], the displacement $\Delta\mathbf{p}$ is approximated by

$$\Delta\mathbf{p}\approx-2[(J^\top J)^{-1}J^\top]\mathscr{E}(\mathbf{p}_t;I_t), \tag{7}$$

where $J=\mathbf{J}(\mathbf{p}_0;I_0)+\mathbf{J}(\mathbf{p}_t;I_t)$, and then the parameter is updated by

$$\mathbf{p}_t\leftarrow\mathbf{p}_t+\Delta\mathbf{p}. \tag{8}$$

The updating is iterated until convergence or maximum iterations reached.

From our observation in experiments, the motion parameter $\mathbf{p}_t$ acquired above usually satisfies the constraints defined in problem (2). Nevertheless, we propose a simple yet effective approach to recompute the warping once the acquired $\mathbf{p}_t$ violates the constraints. Denote $\Omega_t=\{\mathbf{q}|\mathbf{b}\succcurlyeq\mathbf{q}-\mathbf{p}_{t-1}\succcurlyeq-\mathbf{b}\}$ the valid solution space. We sample uniformly $N_s=2500$ candidate solutions $\mathbf{q}_i$ $(1\leq i\leq N_s)$ from Ω_t, and choose the one with minimum dissimilarity

$$\mathbf{p}_t\leftarrow\arg\min_{\mathbf{q}_i}\mathscr{E}_2(\mathbf{q}_i,I_t),\quad 1\leq i\leq N_s. \tag{9}$$

C. Occlusion detection

The occlusion detection strategy adopted in the AKF algorithm [14] is very simple that employs only the difference of intensities for decision of occlusion. This method tends to reject the whole template as outlier in the case of dramatic environmental changes. In [36], the authors propose to discover occlusions through morphological operations on

an occlusion map. However, the approach still does not distinguish occlusions from some other environmental changes, for example motion blurs.

To improve the robustness against occlusion, we propose a novel method of occlusion detection under two empirical guide lines. First, the appearance changes derived from occlusions are diverse enough to be distinguished from other perturbation factors, such as illumination changes and motion blurs, which usually bring similar disturbances to all pixels. Second, The occluded parts are usually connected and compact regions.

On the basis of the above guide lines, we construct current difference image D as $D(\mathbf{x}) = |I_t(\varphi(\mathbf{x};\mathbf{p}_t)) - T(\mathbf{x})|$, and then search occlusions using two criterions as follows.

Diversity criterion. We first compute the mean $\mu(D)$ and the standard deviation $\sigma(D)$ of the difference image D. Obviously, low $\sigma(D)$ indicates less diversity in the difference image. We determine that there is no occlusion if $\sigma(D)/\mu(D) < \theta_0$, where θ_0 is a pre-defined tolerance of the diversity. Otherwise, we go to the further judgement according to the spatial criterion. We set $\theta_0 = 0.8$ throughout our experiments.

Spatial criterion. After binarization on the difference image D, we apply morphological operations to remove the small areas and fill the small hole between the regions. We calculate two attributes $(a_1(R), a_2(R))$ for each connected region R, where $a_1(R)$ denotes the area of region R, and $a_2(R)$ the area of the minimum convex polygon containing region R. We conclude there is an occlusion if $a_1(R)/|D| > \theta_1$ and $a_1(R)/a_2(R) > \theta_2$, where θ_1 and θ_2 are two pre-defined thresholds. The first inequation is to filter out too small regions, and the second to filter out too sparse regions. We set $\theta_1 = 0.1$ and $\theta_2 = 0.5$ throughout our experiments.

To exclude occluded parts from template matching and updating, we directly set the confidence to zero for all occluded pixels.

V. Template updating with Kalman filtering

There have been a few methods [32], [14], [24] utilizing Kalman filters to adapt template for changes. Our approach is inspired by these methods, but differs in the control-input model we employed to capture environmental changes and hence improve the robustness.

A. Kalman filtering

Denote $\mathbf{y}_t$ and $\mathbf{z}_t$ the vectorized state estimation and observation, respectively, of the template intensity at time t. We define the models for state prediction and observation taking the control-input model into account

$$\begin{aligned} \mathbf{y}_t &= A_t \mathbf{y}_{t-1} + B_t \mathbf{u}_t + \mathbf{w}_t, \\ \mathbf{z}_t &= H_t \mathbf{y}_t + \mathbf{v}_t, \end{aligned} \tag{10}$$

where A_t is the state transition matrix which is applied to the previous state $\mathbf{y}_{t-1}$, B_t is the control-input model which is applied to the control vector $\mathbf{u}_t$, H_t is the observation matrix which maps the true state space into the observed space, $\mathbf{w}_t$ and $\mathbf{v}_t$ are the state noise and the observation noise respectively. As common in Kalman filtering, $\mathbf{w}_t$ and $\mathbf{v}_t$ are assumed to be Gaussian with zero means and covariances Q_t and L_t respectively.

In what follows, the notation $\hat{\mathbf{y}}_{t|t'}$ represents the estimate of $\mathbf{y}$ at time t given observations up to time $t' \le t$, and $P_{t|t'}$ the corresponding error covariance. The tracking process consists of the following three distinct phases.

Prediction. We first compute the *priori* state estimation and covariance:

$$\begin{aligned} \hat{\mathbf{y}}_{t|t-1} &= A_t \hat{\mathbf{y}}_{t-1|t-1} + B_{t-1} \mathbf{u}_{t-1}, \\ P_{t|t-1} &= A_t P_{t-1|t-1} A_t^\top + Q_t. \end{aligned} \tag{11}$$

Measuring. To obtain observation of the filters, the previous template $\hat{\mathbf{y}}_{t-1|t-1}$ is matched with the current frame, of which the matching algorithm is discussed detailedly in Sec. IV. The optimal matching result $I_t(\varphi(\mathbf{x};\mathbf{p}_t))$ is used as the observation $\mathbf{z}_t$. The measurement residual and covariance are therefore computed:

$$\begin{aligned} \mathbf{r}_t &= \mathbf{z}_t - H_t \hat{\mathbf{y}}_{t|t-1}, \\ S_t &= H_t P_{t|t-1} H_t^\top + L_t, \end{aligned} \tag{12}$$

Updating. We subsequently update the *posteriori* state estimation and covariance:

$$\begin{aligned} \hat{\mathbf{y}}_{t|t} &= \hat{\mathbf{y}}_{t|t-1} + K_t \mathbf{r}_t, \\ P_{t|t} &= (\mathbf{I} - K_t H_t) P_{t|t-1}, \end{aligned} \tag{13}$$

where the optimal Kalman gain $K_t = P_{t|t-1} H_t^\top S_t^{-1}$, and $\mathbf{I}$ denotes the identity matrix.

It is important for practical implementation of the Kalman Filter to get a good estimate of the model matrices and the noise covariance matrices. However, we lack prior knowledge to utilize off-line learning methods to acquire these matrices. Therefore, we estimate these matrices using on-line learning under some reasonable assumptions:

1) Despite of extrinsic perturbations, the object itself keeps unchanged and can be directly observed. It implies simple models for state transition and observation such that $A_t = \mathbf{I}$ and $H_t = \mathbf{I}$.
2) One promising and practical approach to learn the noise covariance matrices Q_t and L_t is the *auto-covariance least squares* (ALS) technique that uses the time-lagged auto-covariances of routine operating data to estimate the covariances [37]. To reduce both the computational complexity and the dependence on training data, we reduce the noise covariance matrices Q_t and L_t to diagonal matrices under the assumption that the noises of the pixels are independent from each other.
3) The control-input model is introduced to model environmental changes, which is discussed detailedly in Sec. V-B.

B. The control-input model

The control-input model is essential for our method to fight against drastic environmental perturbations. In general, it is hard to know the exact control-input model in advance

in the template tracking task. In this section, we propose an effective method of construction of the control-input model.

The control-input model is approximated according to the probability of the intensity co-occurrence between the pixels. In particular, the control matrix B_t is built as

$$B_t(i,j) = \frac{1}{k} \sum_{m=t-k+1}^{t} c_m(i,j), \tag{14}$$

where $B_t(i,j)$ denotes the element at the i-th row and th j-th column of the control matrix B_t, k controls the size of the window used for computation. The co-occurrence function is defined as

$$c_m(i,j) = \begin{cases} 1 & \text{if } \mathbf{y}_m(i) = \mathbf{y}_m(j), \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

where $\mathbf{y}_m(i)$ and $\mathbf{y}_m(j)$ denote the intensities of the i-th and j-th pixels respectively at time m. The motivation behind this approximation is that pixels with similar intensities tend to hold similar reactions to the input. After the control matrix B_t is built, it is necessary to be normalized as a row stochastic matrix.

The initial control matrix B_0 is built according to the initial template $\mathbf{y}_0$. For computational simplicity, once B_t is computed, we fix $B_t = B_{t+1} = ... = B_{t+k-1}$ until B_{t+k} is updated next time. We set $k = 20$ throughout our experiments.

The environmental input is changing over time, and is hard to be known in advance. After the posteriori estimate $\hat{\mathbf{y}}_{t|t}$ is acquired, we approximate the input $\mathbf{u}_t$ to minimize the squared error between the previous template and the current estimate

$$\mathbf{u}_t^* = \arg\min_{\mathbf{u}_t} ||\hat{\mathbf{y}}_{t|t} - \hat{\mathbf{y}}_{t-1|t-1} - B_t\mathbf{u}_t||_2^2. \tag{16}$$

This optimization is obviously a linear optimization, and can be easily solved. The approximated input $\mathbf{u}_t$, as well as the control matrix B_t, is further used to predict the priori state estimation at time $t+1$.

VI. Experiments

A. Baselines and Benchmarks

In this section, we report experimental results of the proposed CCM algorithm in comparison with eight state-of-the-art baselines, including Struck [5], IC [23], ESM [10], AKF [24], SCV [15], GO-ESM [12], TLD [16] and GPF [17].

Among these algorithms, IC and ESM adopt the classical SSD similarity measure and hold a fixed template without any updating, AKF employs a Kalman filter to adapt the template for changes, SCV and GO-ESM introduce robust similarity measures to replace SSD. Different from the above algorithms based on deterministic optimization, TLD and GPF fall into the probabilistic framework. In particular, Struck is a keypoint-based algorithm which formulates transformation estimation based on keypoint correspondence.

For a thorough evaluation, we report experimental results on two popular benchmarks, UCSB [21] and TMT [22].

- **UCSB:** The dataset comprises 96 video streams displaying six differently textured planar targets with a total of 6,889 frames, featuring geometric distortions (panning, zoom, tilting, rotation), nine levels of motion blur, as well as different lighting conditions, with all frames affected by natural amounts of noise.
- **TMT:** The dataset consists of image sequences of manipulation task recorded by a human user and a robot arm. It contains 109 image sequences with totally 70,592 frames. The recorded videos were grouped under two broad categories: *Oriented Motion Tasks* and *Composite Motion Tasks*. Each oriented motion task refers to one or more highly structured geometric transformations, including zoom, tilting, rotation, translation and occlusion.

All videos come with (semi-)manually annotated ground-truth across all frames. The standard overlap criteria of PASCAL VOC [38] is applied to evaluation in following experiments. Denote R_t^{G} and R_t^{T} the ground-truth region and the tracked region of the object at frame t, respectively. The tracking accuracy at frame t is computed as

$$p_t = \frac{|R_t^{\mathrm{G}} \cap R_t^{\mathrm{T}}|}{|R_t^{\mathrm{G}} \cup R_t^{\mathrm{T}}|}, \tag{17}$$

where $|\cdot|$ denotes the area of a region. The tracking accuracy of a video and hence a dataset is further obtained by average.

B. Experimental results and analysis

In this section, we report tracking performance of the proposed CCM algorithm in comparison with the baseline algorithms. The average tracking accuracy corresponding to each video category is summarized in Tables I (UCSB) and II (TMT).

It is observed that the proposed CCM algorithm outperforms all the baselines remarkably on both datasets. In fact, CCM achieves the best or nearly best tracking performances in almost all video categories, and it exhibits high robustness against not only extreme pose changes but also heavy environmental perturbations. Some baselines provide comparable results to the proposed CCM algorithm on specific categories of videos:

- IC, ESM and AKF exhibit high tracking accuracy in presence of significant pose change, but are very sensitive to illumination change, occlusion and motion blur;
- SCV gains high robustness to illumination by introducing new similarity measure, but it still suffers from occlusion and motion blur;
- The gradient orientation feature brings GO-ESM high robustness to illumination change and partial occlusion, but makes it extremely sensitive to motion blur;
- TLD is able to roughly capture the object in most scenarios, but it fails to get an accurate motion estimation;
- GPF illustrates high robustness against both illumination change and motion blur, at the cost of relatively low tracking accuracy in presence of drastic pose change;
- As a keypoint-based approach, Struck is robust to partial occlusion by design, but is sensitive to extreme pose

TABLE I

AVERAGE TRACKING ACCURACY (±STANDARD DEVIATION) ON THE UCSB DATASET, WHERE BOLD FONT INDICATES THE BEST ACCURACY.

Motion task	Struck [5]	IC [23]	ESM [10]	AKF [24]	SCV [15]	TLD [16]	GPF [17]	GO-ESM [12]	CCM
panning (6)	0.84±0.25	0.29±0.25	0.68±0.31	0.26±0.37	0.71±0.34	0.79±0.13	0.90±0.06	0.35±0.28	**0.92±0.12**
tilting (6)	0.73±0.41	0.82±0.30	0.90±0.19	0.90±0.19	0.90±0.18	0.62±0.38	0.73±0.32	0.90±0.18	**0.91±0.18**
rotation (6)	0.65±0.33	0.74±0.21	**0.80±0.14**	**0.80±0.14**	**0.80±0.14**	0.65±0.18	0.79±0.15	0.79±0.15	**0.80±0.14**
zoom (6)	0.73±0.34	0.73±0.29	**0.92±0.07**	0.82±0.19	**0.92±0.07**	0.77±0.21	0.87±0.11	0.88±0.11	**0.92±0.07**
lighting (12)	0.80±0.33	0.68±0.39	0.83±0.21	0.91±0.13	**0.98±0.02**	0.58±0.42	0.90±0.12	**0.98±0.02**	**0.98±0.02**
blur (54)	0.40±0.41	0.29±0.37	0.43±0.40	0.23±0.36	0.47±0.43	0.65±0.33	0.81±0.14	0.36±0.31	**0.85±0.14**
unconstrained (6)	0.36±0.34	0.07±0.22	0.16±0.27	0.27±0.22	0.07±0.24	0.33±0.34	**0.42±0.38**	0.12±0.21	0.32±0.34
Total (96)	0.53±0.41	0.41±0.34	0.56±0.31	0.44±0.29	0.60±0.31	0.64±0.32	0.80±0.16	0.51±0.38	**0.84±0.15**

TABLE II

AVERAGE TRACKING ACCURACY ON THE TMT DATASET.

object	variation	Struck [5]	IC [23]	ESM [10]	AKF [24]	SCV [15]	TLD [16]	GPF [17]	GO-ESM [12]	CCM
bookI	tilting(12)	0.76±0.37	0.94±0.13	0.98±0.03	0.75±0.26	**0.99±0.02**	0.67±0.24	0.86±0.08	0.98±0.03	**0.99±0.02**
bookII	zoom (13)	0.86±0.24	**0.99±0.01**	**0.99±0.01**	0.96±0.05	**0.99±0.01**	0.74±0.14	0.89±0.06	**0.99±0.01**	**0.99±0.01**
bookIII	occlusion (11)	0.83±0.23	0.43±0.46	0.55±0.43	0.49±0.41	0.55±0.46	0.69±0.21	0.51±0.48	0.84±0.21	**0.85±0.19**
cereal	rotation (13)	0.70±0.44	0.63±0.44	0.82±0.28	0.66±0.39	0.87±0.20	0.64±0.22	0.82±0.25	0.40±0.32	**0.90±0.14**
juice	rotation (13)	0.58±0.43	0.59±0.44	0.78±0.31	0.59±0.36	0.79±0.31	0.59±0.25	0.80±0.28	0.44±0.38	**0.83±0.24**
mugI	translation (13)	0.89±0.13	0.85±0.19	0.93±0.10	0.76±0.28	**0.95±0.07**	0.71± 0.14	0.85±0.16	0.93±0.09	**0.95±0.07**
mugII	tilting (13)	0.70±0.34	0.45±0.47	0.63±0.39	0.47±0.44	0.68±0.36	0.67±0.20	**0.71±0.30**	**0.71±0.30**	**0.71±0.34**
mugIII	rotation (13)	0.75±0.30	0.61±0.37	0.76±0.29	0.56±0.39	0.89±0.14	0.67±0.21	0.79±0.26	0.83±0.21	**0.90±0.14**
Composite	unconstrained (8)	0.48±0.46	0.63±0.35	0.79±0.23	0.63±0.34	**0.92±0.06**	0.57±0.22	0.63±0.25	0.86±0.16	0.90±0.08
Total	- (109)	0.73±0.32	0.68±0.32	0.80±0.23	0.65±0.32	0.85±0.18	0.66±0.20	0.77±0.21	0.77±0.23	**0.89±0.15**

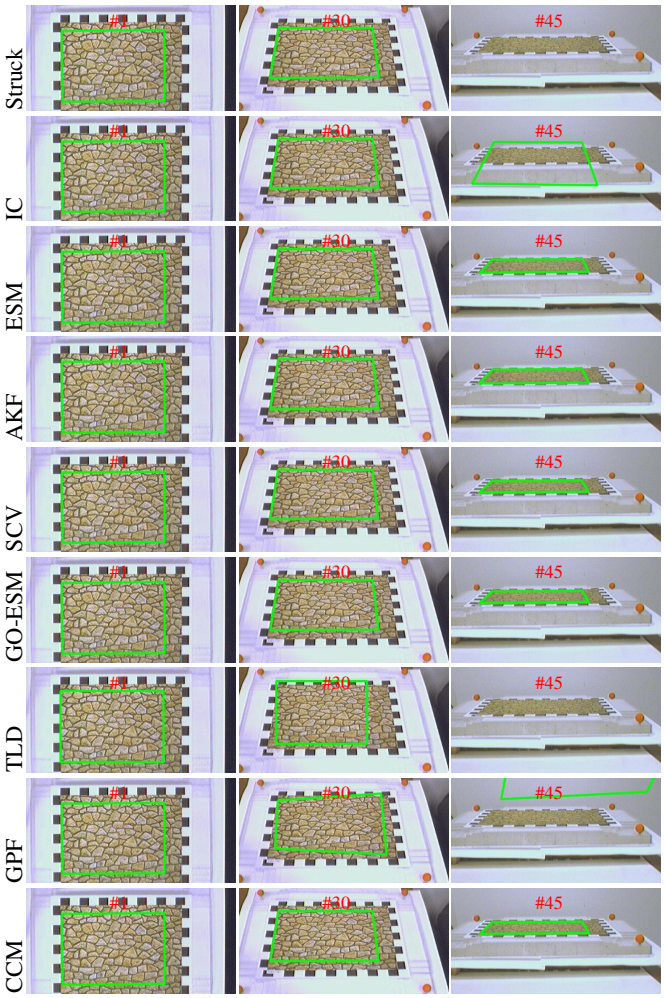

Fig. 2. Examples of tilting a picture with repeat patterns.

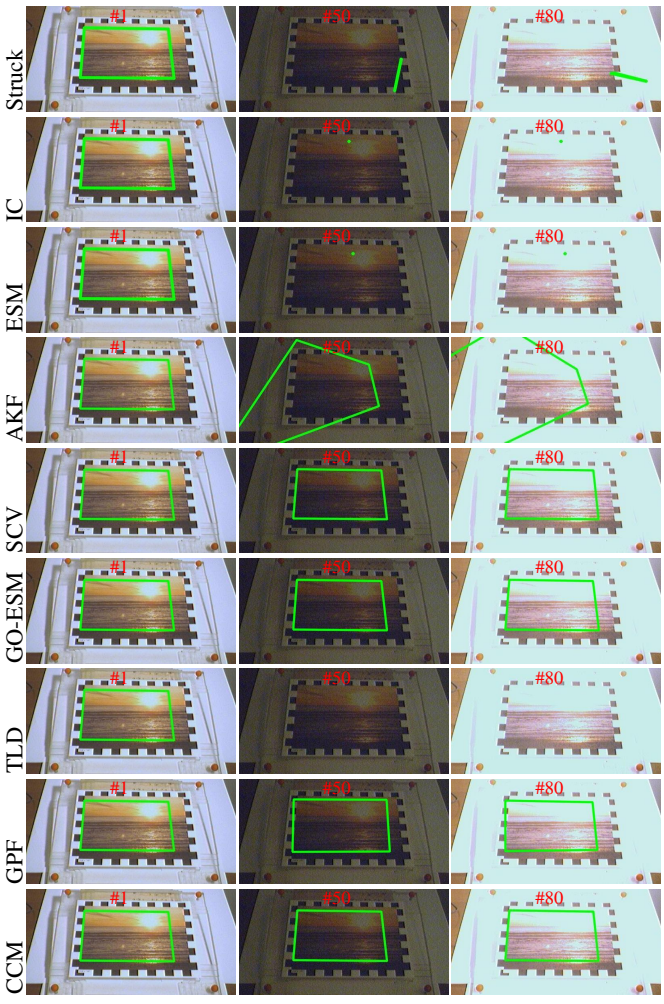

Fig. 3. Examples of drastic and dynamic lighting change of the *sunset* picture with weak texture.

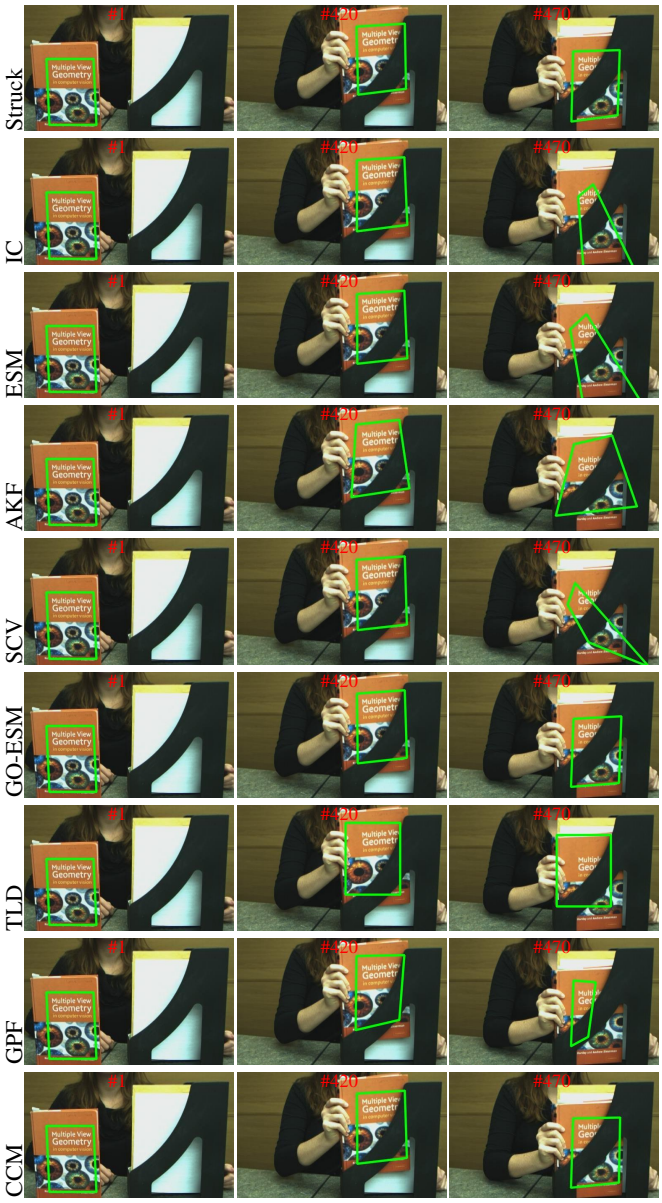

Fig. 4. Example results under occlusion.

TABLE III
AVERAGE COMPUTATIONAL TIME (SECOND) PER FRAME OF THE ALGORITHMS.

Alg.	Struck [5]	IC [23]	ESM [10]	AKF [24]	SCV [15]	TLD [16]	GPF [17]	GO-ESM [12]	CCM
UCSB	0.10	0.21	0.34	0.36	0.18	0.13	0.13	2.12	0.38
TMT	0.07	0.14	0.13	0.19	0.16	0.16	0.12	2.54	0.14

change, illumination change and motion blur.

We also report computational time of the algorithms in Table III. Struck is the most efficient one among these algorithms, and GO-ESM is the most time-consuming one. Despite additional procedures of occlusion detection and template updating are integrated, the proposed CCM algorithm achieves comparable computational efficiency with the original ESM algorithm.

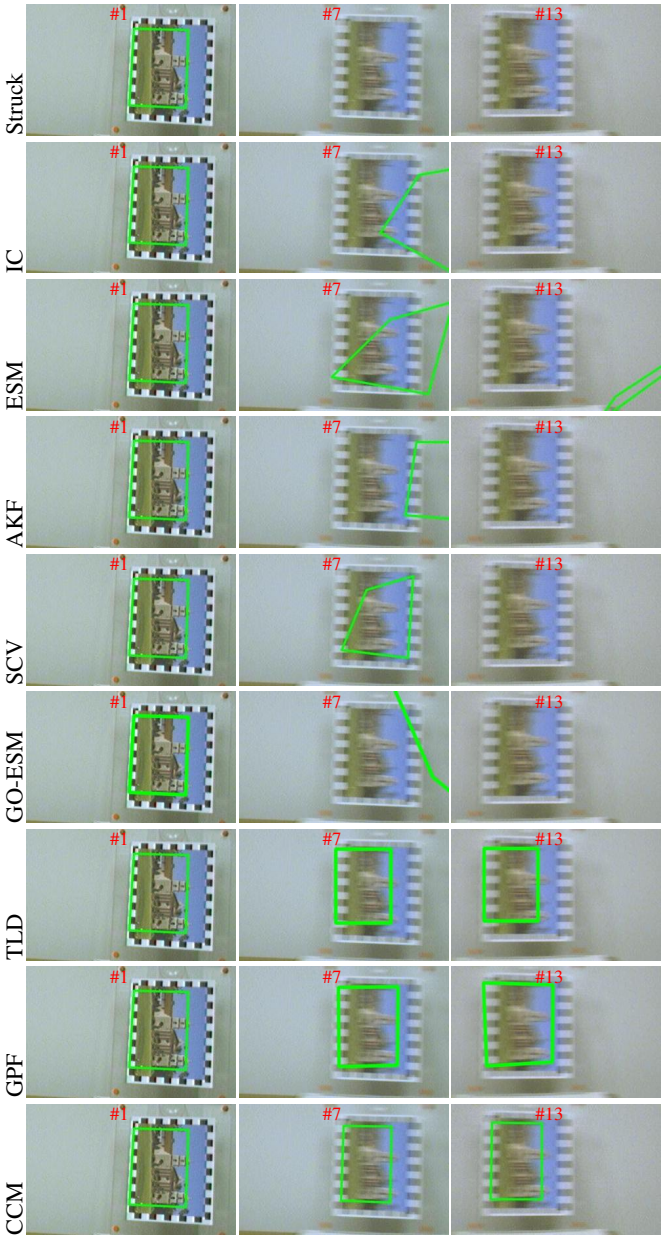

Fig. 5. Examples of motion blur of a picture.

Figures 2 to 5 illustrate several representative examples of various types of intrinsic and extrinsic variations provided by the proposed CCM algorithm in comparison with other algorithms:

- **Tilting.** Example of tilting of a picture with repeat patterns is shown in Fig. 2. It reveals that ESM and its derivations (AKF, SCV, GO-ESM and CCM)[1] are very robust to tilting while Struck, TLD and GPF fail to catch the object in presence of extreme tilting.
- **Lighting.** Fig. 3 shows examples of drastic and dynamic lighting change of the sunset picture with very weak texture. SCV, GO-ESM, GPF and the proposed CCM exhibit high robustness against lighting change where

[1]These algorithms adopt the ESM algorithm for energy minimization.

other algorithms fail to capture the target.

- **Occlusion.** Fig. 4 presents tracking results in presence of partial occlusion. As typical template-based algorithms without tacking occlusion into account, IC, ESM, SCV and GPF naturally suffer from partial occlusion. Although a special mechanism for occlusion is adopted, AKF is still unable to provide accurate tracking results. With the assistance of an online-learned detector, TLD roughly captures the object but with a remarkable drift. The proposed CCM, as well as GO-ESM and Struck, provides more accurate tracking results under partial occlusion.
- **Motion blur.** In Fig. 5, Stuck is very sensitive to motion blur because less reliable keypoints are detected in this case. Without proper updating of the template, IC, ESM, AKF, SCV and GO-ESM also lose the target due to drastic appearance change. In contrast, TLD, GPF and the proposed CCM provide more accurate results across all frames.

VII. Conclusion

In this paper, we proposed a novel constrained confidence matching algorithm for planar object tracking aiming to improve the tracking performance. We employ a control-input model in the Kalman filter to capture the environmental changes, and develop an accurate occlusion detector to reject occlusion from motion estimating and template updating. Experimental results reveal that, the proposed approach gains accurate and robust tracking performance against various environmental variations, and outperforms recent state-of-the-art algorithms.

ACKNOWLEDGMENT

This work is supported by China National Key Research and Development Plan (Grant No. 2016YFB1001200), the National Nature Science Foundation of China (nos. 61673048, 61672088, 61671048, 61528204 and 61472028), the Fundamental Research Funds for the Central universities (2017JBZ108), and a Research Grant from HiScene Information Technologies.

References

[1] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *ICRA*, 2013, pp. 3748–3754.

[2] H. Lategahn, A. Geiger, and B. Kitt, "Visual SLAM for autonomous ground vehicles," in *ICRA*, 2011, pp. 1732–1737.

[3] A. Makadia, A. Patterson, and K. Daniilidis, "Fully automatic registration of 3d point clouds," in *CVPR*, 2006, pp. 1297–1304.

[4] L. Torresani, A. Hertzmann, and C. Bregler, "Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors," *PAMI*, vol. 30, no. 5, pp. 878–892, 2008.

[5] S. Hare, A. Saffari, and P. Torr, "Efficient online structured output learning for keypoing-based object tracking," in *CVPR*, 2012, pp. 1894–1901.

[6] M. Özuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast keypoint recognition using random ferns," *PAMI*, vol. 32, no. 3, pp. 448–461, 2010.

[7] T. Wang and H. Ling, "Gracker: A graph-based planar object tracker," *PAMI*, 2017. [Online]. Available: https://doi.org/10.1109/TPAMI.2017.2716350

[8] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, 1981, pp. 674–679.

[9] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *IJCV*, vol. 56, no. 3, pp. 221–255, 2004.

[10] E. Malis, "Improving vision-based control using efficient second-order minimization techniques," in *ICRA*, 2004, pp. 1843–1848.

[11] S. Holzer, M. Pollefeys, S. llic, D. J. Tan, and N. Navab, "Online learning of linear predictors for real-time tracking," in *ECCV*, 2012, pp. 470–483.

[12] L. Chen, F. Zhou, Y. Shen, X. Tian, H. Ling, and Y. Chen, "Illumination insensitive efficient second-order minimization for planar object tracking," in *ICRA*, 2017, pp. 751–757.

[13] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *PAMI*, vol. 26, no. 6, pp. 810–815, 2004.

[14] H. T. Nguyen, M. Worring, and R. van den Boomgaard, "Occlusion robust adaptive template tracking," in *ICCV*, 2001, pp. 678–683.

[15] R. Richa, M. de Souza, G. G. Scandaroli, E. Comunello, and A. von Wangenheim, "Direct visual tracking under extreme illumination variations using the sum of conditional variance," in *ICIP*, 2014, pp. 373–377.

[16] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *PAMI*, vol. 34, no. 7, pp. 1409–1422, 2012.

[17] J. Kwon, H. S. Lee, F. C. Park, and K. M. Lee, "A geometric particle filter for template-based visual tracking," *PAMI*, vol. 36, no. 4, pp. 625–643, 2014.

[18] R. Richa, R. Sznitman, R. Taylor, and G. Hager, "Visual tracking using the sum of conditional variance," in *IROS*, 2011, pp. 2953–2958.

[19] G. G. Scandaroli, M. Meilland, and R. Richa, "Improving ncc-based direct visual tracking," in *ECCV*, 2012, pp. 442–455.

[20] A. Dame and É. Marchand, "Accurate real-time tracking using mutual information," in *ISMAR*, 2010, pp. 47–56.

[21] S. Gauglitz, T. Höllerer, and M. Turk, "Evaluation of interest point detectors and feature descriptors for visual tracking," *IJCV*, vol. 94, no. 3, pp. 335–360, 2011.

[22] A. Roy, X. Zhang, N. Wolleb, C. P. Quintero, and M. Jägersand, "Tracking benchmark and evaluation for manipulation tasks," in *ICRA*, 2015, pp. 2448–2453.

[23] G. Hager and P. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *PAMI*, vol. 20, no. 10, pp. 1025–1039, 1998.

[24] H. T. Nguyen and A. Smeulders, "Fast occluded object tracking by a robust appearance filter," *PAMI*, vol. 26, no. 8, pp. 1099–1104, 2004.

[25] V. Lepetit and P. Fua, "Monocular model-based 3d tracking of rigid objects: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 1, no. 1, 2005.

[26] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM CSUR*, vol. 38, no. 4, p. 13, 2006.

[27] P. Besl and N. McKay, "A method for registration of 3-d shape," *PAMI*, vol. 14, no. 2, pp. 239–256, 1992.

[28] M. Gleicher, "Projective registration with difference decomposition," in *CVPR*, 1997, pp. 331–337.

[29] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," in *ECCV*, 1998, pp. 484–498.

[30] M. Turk, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[31] M. Black and A. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *IJCV*, vol. 26, no. 1, pp. 63–84, 1998.

[32] F. Dellaert, S. Thrun, and C. Thorpe, "Jacobian images of super-resolved texture maps for model-based motion estimation and tracking," in *WACV*, 1998, pp. 2–7.

[33] D. A. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *IJCV*, vol. 77, no. 1-3, pp. 125–141, 2008.

[34] F. Wang and B. C. Vemuri, "Non-rigid multi-modal image registration using cross-cumulative residual entropy," *IJCV*, vol. 74, no. 2, pp. 201–215, 2007.

[35] D. J. Tan and S. Ilic, "Multi-forest tracker: A chameleon in tracking," in *CVPR*, 2014, pp. 1202–1209.

[36] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient l1 tracker with occlusion detection," in *CVPR*, 2011, pp. 1257–1264.

[37] M. Rajamania and J. Rawlingsb, "Estimation of the disturbance structure from data using semidefinite programming and optimal weighting," *Automatica*, vol. 45, no. 1, pp. 142–148, 2009.

[38] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *IJCV*, vol. 88, no. 2, pp. 303–338, 2010.