6-DOF Direct Homography Tracking with Extended Kalman Filter

Hyowon Ha, François Rameau, and In So Kweon(✉)

Robotics and Computer Vision Laboratory, KAIST, Daejeon, South Korea
iskweon@kaist.ac.kr

Abstract. This paper considers a robust direct homography tracking that takes advantage of the known intrinsic parameters of the camera to estimate its pose in real scale, to speed-up the convergence, and to drastically increase the robustness of the tracking. Indeed, our new formulation for direct homography tracking allows us to explicitly solve a 6 Degrees Of Freedom (DOF) rigid transformation between the plane and the camera. Furthermore, it simplifies the integration of the Extended Kalman Filter (EKF) which allows us to increase the computational speed and deal with large motions. For the sake of robustness, our approach also includes a pyramidal optimization using an Enhanced Correlation Coefficient (ECC) based objective function. The experiments show the high efficiency of our approach against state of the art methods and under challenging conditions.

Keywords: ECC · Homography tracking · Pose estimation · EKF

1 Introduction

In the past decade, planar homography tracking has been extensively studied in the field of computer vision since it is an essential tool for a large number of applications, such as visual servoing, robotic navigation, augmented reality and more.

The direct homography tracking pipeline is straightforward; a known image template is tracked along a video sequence by iteratively solving a parametric image alignment problem which minimize the photometric difference between the template and the current image. The estimated parameters from the previous frame are then utilized as an initial guess for the current one. This process is repeated for every new frame in the sequence. This mapping between the coordinates of both images requires an appropriate geometric transformation. For planar tracking using a pinhole camera, the perspective homography is considered to be the most suitable model for general cases [1].

This strategy is very efficient but relies on multiple assumptions, such as a small and smooth inter-frame displacement. This first hypothesis is violated when fast motions are performed, which limits the possible uses for this technique in many practical applications. Hence, many researches have focused on

T. Bräunl et al. (Eds.): PSIVT 2015, LNCS 9431, pp. 447–460, 2016.
DOI: 10.1007/978-3-319-29451-3_36

more efficient objective functions based on different performance criterions like Mutual Information (MI) [2], NCC [3], etc. The goal of the previously mentioned approaches is to increase the robustness against illumination changes and the range of convergence in order to handle larger motions than basic SSD approaches [4]. Nonetheless, a large overlap - of the tracked plane between two consecutive images - is still required to ensure correct tracking. The interested reader can check the results obtained with different direct homography tracking methods under fast motions in [3]. The first attempt specifically designed to deal with fast motion is probably the work of Park *et al.* [8], where the well-known Efficient Second Order Minimization (ESM) tracking [4] is modified to deal with strongly blurred images.

Also, non-direct approaches using sparse features such as edges [5] or points [6] exist but are usually very sensitive to motion blur and strong changes in scale and appearance. Direct approaches tend to be more robust in such challenging conditions. However the feature-based methods are very useful for the re-detection of the target and less prone to drift. In [7], the authors combine the advantages from both approaches in a single hybrid scheme. In this paper, we exclusively focused on the direct tracking approaches.

From the literatures, we acknowledge two facts. First, most of the existing methods focused on uncalibrated camera configurations which consist of the resolution of an 8 degrees of freedom problem in order to solve a homography (up to scale). When the camera pose is needed, the extrinsic parameters are extracted afterwards using the intrinsic parameters of the camera [2]. This homography decomposition does not ensure the orthogonality of the rotation matrix which has to be enforce afterwards. This leads to a slightly biased estimation of the motion. In this work, we propose to include the computation of the extrinsic parameters directly in the tracking process through a reformulation of the problem. If the size of the template is known, the real scale pose estimation can be determined by our method. Moreover, explicitly solving the orientation and position of the camera leads to a more constrained problem less prone to divergence than usual approaches.

Our second observation is that the existing methods are purely deterministic and remains very sensitive to fast motions. In this paper, we propose to include a probabilistic stage in the tracking process. Indeed, our reformulation of the tracking problem allows for the use of the EKF [9,10] to predict the next pose of the camera. Therefore, this prediction can be utilized as an initialization for the next frame. With this approach, we are able to deal with larger interframe motions than conventional methods since the predicted initial parameters are closer to the optimal solution. This procedure also strongly increases the convergence speed of the optimization step. To our best knowledge, it is the first attempt to fuse the EKF to direct homography tracking, while the EKF has been intensively adopted to many non-direct tracking methods [5,11].

Another strong assumption for direct homography tracking is the lightness constancy. A large number of works dedicated to this particular problem are available. For instance, in [12], Silveira *et al.* cope with generic illumination changes using an efficient illumination model.

Although it is not the main point of this paper, our approach also takes the local and global illumination changes into consideration, thanks to a robust objective function based on ECC [1]. The range of convergence as well as the speed of the method have also been improved by a pyramidal optimization scheme. However, it is clear that our approach is compatible with any performance criterion. The advantages offered by our method are underlined through multiple experiments where the tracking accuracy, robustness and speed are evaluated.

This paper is organized in the following manner: in the Sect. 2 we describe both the reformulation of the homography under a 6DOF problem and the ECC-based image alignment process. The next section is dedicated to the integration of the Extended Kalman Filter in the tracking scheme. In the Sect. 4, we propose a large number of results demonstrating the accuracy and the speed of our method. Finally this paper ends with a short conclusion.

2 ECC-Based 6-DOF Direct Homography Tracking

In this section, our ECC-based 6-DOF direct homography tracking algorithm is explained in twofold. Firstly, our homography from a 6-DOF pose is explained. Compared to conventional approaches, our homography is modeled using the 6-DOF pose of the camera thanks to its known intrinsic parameters. This new formulation enhances the convergence of the optimization and allows us to apply the EKF. Secondly, our ECC-based tracking algorithm is described in detail. We model the homography-based tracking of a planar object as an ECC-based nonlinear least squares problem with 6 unknown parameters which can be solved using a gradient descent approach.

2.1 Homography from a 6-DOF Pose

In computer vision, projective homography is often referred to as a transformation between two images of the same planar object. It is geometrically modeled by the normal direction of the plane, the intrinsic parameters and the poses of the cameras. Practically, it is often utilized to estimate the relative pose of the camera w.r.t. a plane. If we consider two cameras as a and b, the corresponding image points $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ observed by a and b follow this homographic relationship:

$$\boldsymbol{x}_a \sim \mathbf{K}_a \cdot \mathbf{H}_{b \rightarrow a} \cdot \mathbf{K}_b^{-1} \cdot \boldsymbol{x}_b, \tag{1}$$

where $\mathbf{H}_{b \rightarrow a}$ is a 3×3 homography transformation matrix from b to a and $\mathbf{K}_a$, $\mathbf{K}_b$ are the intrinsic camera matrices of a and b.

In this work, we focus on a homography-based tracking scenario where we have a single image for b (*template*) and a series of images for a (*target*). Therefore, we use abbreviated notations tgt (*target*) and tmp (*template*) in place of a and b respectively (Fig. 1). We design our world coordinate to be centered at the planar object of which the plane equation is $Z = 0$. If an image of

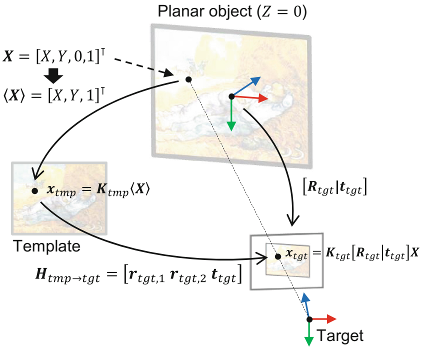
Planar object (Z = 0)
$\boldsymbol{X} = [X, Y, 0,1]^\top$
$\langle\boldsymbol{X}\rangle = [X, Y, 1]^\top$
$\boldsymbol{x}_{tmp} = \boldsymbol{K}_{tmp}\langle\boldsymbol{X}\rangle$
Template
$[\boldsymbol{R}_{tgt}|\boldsymbol{t}_{tgt}]$
$\boldsymbol{x}_{tgt} = \boldsymbol{K}_{tgt}[\boldsymbol{R}_{tgt}|\boldsymbol{t}_{tgt}]\boldsymbol{X}$
$\boldsymbol{H}_{tmp\rightarrow tgt} = [\boldsymbol{r}_{tgt,1}\ \boldsymbol{r}_{tgt,2}\ \boldsymbol{t}_{tgt}]$
Target

Fig. 1. Our homography relationship between the planar object, template image and the target image.

the planar object is captured by the *target* camera with an intrinsic camera matrix $\mathbf{K}_{tgt}$ and a 6-DOF pose represented as $[\mathbf{R}_{tmp}, \mathbf{t}_{tmp}]$, an arbitrary 3D point $\mathbf{X} = [X, Y, 0, 1]^\top$ on the planar object can be projected onto the *target* image as $\mathbf{x}_{tgt} = \mathbf{K}_{tgt}\left[\mathbf{R}_{tgt}|\mathbf{t}_{tgt}\right]\mathbf{X}$. By introducing a deprived form $\langle\mathbf{X}\rangle = [X, Y, 1]^\top$ to reflect the zero z-value, the projection can be simplified as:

$$\mathbf{x}_{tgt} = \mathbf{K}_{tgt}\left[\mathbf{r}_{tgt,1}, \mathbf{r}_{tgt,2}, \mathbf{t}_{tgt}\right]\langle\mathbf{X}\rangle \tag{2}$$

The corresponding point in the *template* image can be calculated by a transformation $\mathbf{K}_{tmp}$ (like an intrinsic camera matrix) so that $\mathbf{x}_{tmp} = \mathbf{K}_{tmp}\langle\mathbf{X}\rangle$ where $\mathbf{K}_{tmp}$ must be predefined w.r.t. the *pixel/mm* scale of the *template* image and the position of the reference coordinate. For instance, if the *template* image consists of $w \times h$ pixels and the actual size of the corresponding rectangle on the planar object is $W \times H$ millimeters, then it can be expressed in the following manner:

$$\mathbf{K}_{tmp} = \begin{bmatrix} \frac{w-1}{W} & 0 & \frac{w-1}{2} \\ 0 & \frac{h-1}{H} & \frac{h-1}{2} \\ 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

thereby enforcing the reference coordinate system to be located at the center of the template image and the pose estimation in metric scale to be achieved.

From Eqs. (1), (2) and (3), the homography from the *template* image to the *target* image, $\mathbf{H}_{tmp\rightarrow tgt}$, can be described simply as follows:

$$\mathbf{H}_{tmp\rightarrow tgt} = \left[\boldsymbol{r}_{tgt,1}\ \ \boldsymbol{r}_{tgt,2}\ \ \boldsymbol{t}_{tgt}\right]. \tag{4}$$

2.2 ECC-based Direct Homography Tracking

ECC [1] is one of the image similarity measures which quantify the similarity between two images. Hence, the alignment of the two images can be achieved by maximizing the ECC between them. Given a pair of pixel intensities $I_{tmp}(\boldsymbol{x}_k)$ and $I_{tgt}(\boldsymbol{y}_k)$ for the template image and the target image (respectively at the image coordinates $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$), the alignment problem consists of finding the transformation vector $\boldsymbol{p}$ mapping $\boldsymbol{y}_k = \boldsymbol{\phi}(\boldsymbol{x}_k;\boldsymbol{p})$. In our case, $\boldsymbol{p}$ is a vector of length 6 composed of the rotation vector $\boldsymbol{r}_{tgt}$ and the translation vector $\boldsymbol{t}_{tgt}$ of the *target* camera. In this paper, the vector for pixels' intensities of the template image ($\boldsymbol{i}_{tmp}$) and the target image ($\boldsymbol{i}_{tgt}$) are defined as:

$$\boldsymbol{i}_{tmp} = \left[I_{tmp}\left(\boldsymbol{x}_1\right), I_{tmp}\left(\boldsymbol{x}_2\right), \cdots, I_{tmp}\left(\boldsymbol{x}_K\right)\right]^\top, \tag{5}$$

$$\boldsymbol{i}_{tgt}(\boldsymbol{p}) = \left[I_{tgt}\left(\boldsymbol{y}_1(\boldsymbol{p})\right), I_{tgt}\left(\boldsymbol{y}_2(\boldsymbol{p})\right), \cdots, I_{tgt}\left(\boldsymbol{y}_K(\boldsymbol{p})\right)\right]^\top. \tag{6}$$

Thus, our ECC-based image alignment problem solving for the optimal vector $\boldsymbol{p}^*$ is defined as follows:

$$\boldsymbol{p}^* = \arg\min_{\boldsymbol{p}} \sum_{k=1}^{K} \left(\frac{I_{tgt}(\boldsymbol{y}_k(\boldsymbol{p})) - \bar{\boldsymbol{i}}_{tgt}}{\left\|\boldsymbol{i}_{tgt} - \bar{\boldsymbol{i}}_{tgt}\right\|} - \frac{I_{tmp}(\boldsymbol{x}_k) - \bar{\boldsymbol{i}}_{tmp}}{\left\|\boldsymbol{i}_{tmp} - \bar{\boldsymbol{i}}_{tmp}\right\|} \right)^2 \tag{7}$$

where $\bar{\boldsymbol{i}}_{tmp}$ and $\bar{\boldsymbol{i}}_{tgt}$ are the mean values of $\boldsymbol{i}_{tmp}$ and $\boldsymbol{i}_{tgt}$, while $\|\cdot\|$ stands for the Euclidean distance.

To solve this non-linear least squares problem, we adopt the Levenberg-Marquardt (LM) [13] algorithm, one of the most famous gradient descent methods. For an efficient use of the LM algorithm, it is essential to compute the Jacobian matrix of the objective function.

Since the image intensity is not a continuous function, we consider its approximation by applying the first-order Taylor expansion for $\boldsymbol{p} = \tilde{\boldsymbol{p}} + \triangle\boldsymbol{p}$ where $\triangle\boldsymbol{p}$ is a vector of perturbations:

$$I_{tgt}(\boldsymbol{y}(\boldsymbol{p})) \approx I_{tgt}(\boldsymbol{y}(\tilde{\boldsymbol{p}})) + \left[\bigtriangledown_y I_{tgt}(\boldsymbol{y}(\tilde{\boldsymbol{p}}))\right]^\top \frac{\partial \boldsymbol{\phi}(\boldsymbol{x};\tilde{\boldsymbol{p}})}{\partial \boldsymbol{p}} \triangle\boldsymbol{p}, \tag{8}$$

where $\bigtriangledown_y I_{tgt}(\boldsymbol{y}(\tilde{\boldsymbol{p}}))$ is the vector of the gradient intensities of the image I_{tgt} at $\boldsymbol{y}(\tilde{\boldsymbol{p}})$ and $\frac{\partial \boldsymbol{\phi}(\boldsymbol{x};\tilde{\boldsymbol{p}})}{\partial \boldsymbol{p}}$ is the Jacobian matrix of the transformation vector mapping $\boldsymbol{\phi}$ with respect to the parameters. Now, we rewrite Eq. (1) with vector normalization:

$$\hat{\boldsymbol{y}} = \hat{\boldsymbol{\phi}}(\boldsymbol{x};\boldsymbol{p}) = \mathbf{K}_{tgt} \cdot \mathbf{H}(\boldsymbol{p}) \cdot \mathbf{K}_{tmp}^{-1} \cdot \boldsymbol{x}, \tag{9}$$

$$\boldsymbol{y} = \boldsymbol{\phi}(\boldsymbol{x};\boldsymbol{p}) = \left[\frac{\hat{y}_1}{\hat{y}_3}, \frac{\hat{y}_2}{\hat{y}_3}\right]^\top, \tag{10}$$

where $\boldsymbol{x}$, $\boldsymbol{y}$ are the coordinates in the *template* and the *target* images, with $\mathbf{K}_{tmp}$, $\mathbf{K}_{tgt}$ as their intrinsic matrices. Here, $\mathbf{H}(\boldsymbol{p})$ and $\mathbf{K}_{tmp}$ are expressed as defined in Eqs. (3) and (4). To consider the normalization of $\hat{\boldsymbol{y}}$ (Eq. (10)), the vector function $\hat{\boldsymbol{\phi}}$ is divided into three scalar functions $\hat{y}_1 = \hat{\phi}_1(\boldsymbol{x};\boldsymbol{p})$, $\hat{y}_2 = \hat{\phi}_2(\boldsymbol{x};\boldsymbol{p})$ and $\hat{y}_3 = \hat{\phi}_3(\boldsymbol{x};\boldsymbol{p})$. Then, the vector mapping function $\boldsymbol{\phi}$ is represented by the two components, $\phi_1(\boldsymbol{x};\boldsymbol{p}) = \hat{\phi}_1(\boldsymbol{x};\boldsymbol{p})/\hat{\phi}_3(\boldsymbol{x};\boldsymbol{p})$ and $\phi_2(\boldsymbol{x};\boldsymbol{p}) = \hat{\phi}_2(\boldsymbol{x};\boldsymbol{p})/\hat{\phi}_3(\boldsymbol{x};\boldsymbol{p})$.

The partial derivatives of $\boldsymbol{\phi}$ with respect to p_i, or the *i-th* element of $\boldsymbol{p}$ where $i = 1, \cdots, 6$, can be calculated as follows:

$$\frac{\partial \boldsymbol{\phi}(\boldsymbol{x})}{\partial p_i} = \left[\frac{\partial \phi_1(\boldsymbol{x})}{\partial p_i}, \frac{\partial \phi_2(\boldsymbol{x})}{\partial p_i} \right]^\top, \tag{11}$$

$$\frac{\partial \phi_1(\boldsymbol{x})}{\partial p_i} = \frac{\partial \hat{\phi}_1(\boldsymbol{x})}{\partial p_i} \left(\frac{1}{\hat{\phi}_3(\boldsymbol{x})} \right) - \frac{\partial \hat{\phi}_3(\boldsymbol{x})}{\partial p_i} \left(\frac{\hat{\phi}_1(\boldsymbol{x})}{\hat{\phi}_3^2(\boldsymbol{x})} \right), \tag{12}$$

$$\frac{\partial \phi_2(\boldsymbol{x})}{\partial p_i} = \frac{\partial \hat{\phi}_2(\boldsymbol{x})}{\partial p_i} \left(\frac{1}{\hat{\phi}_3(\boldsymbol{x})} \right) - \frac{\partial \hat{\phi}_3(\boldsymbol{x})}{\partial p_i} \left(\frac{\hat{\phi}_2(\boldsymbol{x})}{\hat{\phi}_3^2(\boldsymbol{x})} \right), \tag{13}$$

where

$$\begin{bmatrix} \frac{\partial \hat{\phi}_1(\boldsymbol{x})}{\partial p_i} \\ \frac{\partial \hat{\phi}_2(\boldsymbol{x})}{\partial p_i} \\ \frac{\partial \hat{\phi}_3(\boldsymbol{x})}{\partial p_i} \end{bmatrix} = \mathbf{K}_{tgt} \cdot \frac{\partial \mathbf{H}}{\partial p_i} \cdot \mathbf{K}_{tmp}^{-1} \cdot \boldsymbol{x}. \tag{14}$$

Since we consider $\boldsymbol{p} = \left[\boldsymbol{r}_{tgt}^\top, \boldsymbol{t}_{tgt}^\top \right]^\top$, the partial derivative of $\mathbf{H}$ depends on the rotation representation. In this paper, we use the Rodrigues' rotation formula.

Finally, the Jacobian matrix $\mathbf{J}$ for Eq. (7) is computed as:

$$\mathbf{J} = \begin{bmatrix} J_{1,1} & J_{1,2} & \cdots & J_{1,6} \\ J_{2,1} & J_{2,2} & \cdots & J_{2,6} \\ \vdots & \vdots & \ddots & \vdots \\ J_{K,1} & J_{K,2} & \cdots & J_{K,6} \end{bmatrix}, \tag{15}$$

where

$$J_{k,i} = \left[\nabla_y I_{tgt}(\boldsymbol{y}_k) \right]^\top \begin{bmatrix} \frac{\partial \phi_1(\boldsymbol{x})}{\partial p_i} \\ \frac{\partial \phi_2(\boldsymbol{x})}{\partial p_i} \end{bmatrix} / \left\| \boldsymbol{i}_{tgt} - \bar{\boldsymbol{i}}_{tgt} \right\|, \tag{16}$$

by assuming $\bar{\boldsymbol{i}}_{tgt}(\boldsymbol{p}) \approx \bar{\boldsymbol{i}}_{tgt}(\tilde{\boldsymbol{p}})$ and $\left\| \boldsymbol{i}_{tmp}(\boldsymbol{p}) - \bar{\boldsymbol{i}}_{tmp}(\boldsymbol{p}) \right\| \approx \left\| \boldsymbol{i}_{tmp}(\tilde{\boldsymbol{p}}) - \bar{\boldsymbol{i}}_{tmp}(\tilde{\boldsymbol{p}}) \right\|$.

The LM method proceeds through successive iterations from an initial guess $\boldsymbol{p}^{(0)}$ as:

$$\boldsymbol{p}^{(s+1)} = \boldsymbol{p}^{(s)} - \left(\mathbf{J}^\top \mathbf{J} + \lambda \mathrm{diag}(\mathbf{J}^\top \mathbf{J}) \right)^{-1} \mathbf{J}^\top \boldsymbol{f}(\boldsymbol{p}^{(s)}) \tag{17}$$

where $\boldsymbol{f}$ is a function (see Eq. (7)) returning a residual vector of length K, s is the iteration step and λ is the damping factor for the LM method.

3 Integration of Extended Kalman Filter

The Kalman Filter (KF) [14] is a method that uses a series of noisy measurements to produce estimates of the state of a system. However, the basic KF is limited to linear systems while many actual systems (such as the rotational motion model) are inherently non-linear. To overcome this limitation, the Extended Kalman Filter (EKF) [9,10] has been introduced as an extended version of the KF for non-linear systems. To be compatible with non-linearity, the EKF takes advantage of the partial derivatives (Jacobian) of the non-linear system at each time step under the assumption that it is locally linear. Though the EKF is imperfect for estimating an optimal solution due to this hypothesis, it still provides a reliable state prediction with very low computational cost, which is essential for various real-time applications.

We adopt the EKF specifically to provide a well predicted initial pose for our direct homography tracking algorithm instead of using only the previous pose estimation. The EKF consist of three steps; prediction, tracking, and correction. In the prediction step, the EKF produces estimates of the current state variables regarding the camera motion along with their uncertainties. In the tracking step, our ECC-based direct homography tracking algorithm is applied with the predicted pose of the camera, and produces a refined camera pose which we call the measurement. In the correction step, the estimates and the uncertainties are updated based on the new measurement in a weighted averaging manner, where a larger weight is attributed to the estimates with higher certainty.

In this work, we divide the motion model of the camera into two systems: a linear system for the translational motion and a non-linear system for the rotational motion. Even though the two systems can be modelled together in a single EKF scheme, it is more efficient to separate them to reduce the computational cost since they can be regarded as being independent to each other. The details of each model are explained in the following sections. (Please note that the notations in this section are independent of those in Sect. 2.)

3.1 Translational Motion Model

The translational motion can be modeled by the following linear system:

$$\boldsymbol{t}_k = \boldsymbol{t}_{k-1} + \boldsymbol{v}_{k-1}\varDelta t + \frac{1}{2}\boldsymbol{a}_{k-1}(\varDelta t)^2, \tag{18}$$

$$\boldsymbol{v}_k = \boldsymbol{v}_{k-1} + \boldsymbol{a}_{k-1}\varDelta t, \tag{19}$$

$$\boldsymbol{a}_k = \boldsymbol{a}_{k-1}, \tag{20}$$

where the subscript k denotes the time step, $\varDelta t$ is the time interval and $\boldsymbol{t} = [t_x, t_y, t_z]^\top$, $\boldsymbol{v} = [v_x, v_y, v_z]^\top$, $\boldsymbol{a} = [a_x, a_y, a_z]^\top$ are the vectors of the translation, velocity, and acceleration, respectively.

Since it is a linear system, we can apply the basic KF for predicting the current translational motion. The state vector is the concatenation of the translation, velocity and acceleration vectors. Hence, the state transition model can be written as follows:

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{t}_k \\ \boldsymbol{v}_k \\ \boldsymbol{a}_k \end{bmatrix} = \mathbf{F}_k \boldsymbol{x}_{k-1}, \tag{21}$$

where $\boldsymbol{x}_k$ is the state vector of length 9 and $\mathbf{F}_k$ is the 9×9 state transition matrix implying Eqs. (18), (19) and (20). For the sake of clarity, we omit the process noise in the model which is assumed to be a zero mean Gaussian white noise.

At time k, the measurement $\boldsymbol{z}_k$ of the state $\boldsymbol{x}_k$ is modeled as:

$$\boldsymbol{z}_k = \mathbf{H}_k \boldsymbol{x}_k, \tag{22}$$

$$\mathbf{H}_k = \left[\mathbf{I}_{3\times3}\ \mathbf{0}_{3\times3}\ \mathbf{0}_{3\times3}\right], \tag{23}$$

where $\boldsymbol{z}_k$ is the measurement vector of length 3 and $\mathbf{H}_k$ is the 3×9 measurement matrix. Here, $\boldsymbol{z}_k$ is the translation vector estimated by our direct homography tracking which utilizes the predicted translation from Eq. (21) as the initial guess. Again, for simplicity, we do not detail the measurement noise which is also assumed to be zero mean Gaussian white noise.

3.2 Rotational Motion Model

We define the rotational motion model using the quaternions and an angular velocity vector which allow for a closed-form propagation of the rotational motion:

$$\boldsymbol{q}_k = e^{\boldsymbol{\Omega}(\boldsymbol{w}_{k-1})\Delta t}\boldsymbol{q}_{k-1}, \tag{24}$$

$$\boldsymbol{w}_k = \boldsymbol{w}_{k-1}, \tag{25}$$

where k is the time step, $\boldsymbol{q}$ is the quaternion vector of length 4, $\boldsymbol{w} = [w_x, w_y, w_z]^\top$ is the angular velocity vector and $\boldsymbol{\Omega}(\boldsymbol{w})$ is defined as:

$$\boldsymbol{\Omega}(\boldsymbol{w}) = \frac{1}{2}\begin{bmatrix} 0 & w_z & -w_y & w_x \\ -w_z & 0 & w_x & w_y \\ w_y & -w_x & 0 & w_z \\ -w_x & -w_y & -w_z & 0 \end{bmatrix}. \tag{26}$$

Using the power series expansion, Eq. (24) can be reduced to:

$$\boldsymbol{q}_k = \left[\cos\left(|\boldsymbol{w}_{k-1}|\,\Delta t/2\right)\mathbf{I}_{4\times4} + \frac{2}{|\boldsymbol{w}_{k-1}|}\sin\left(|\boldsymbol{w}_{k-1}|\,\Delta t/2\right)\boldsymbol{\Omega}(\boldsymbol{w}_{k-1})\right]\boldsymbol{q}_{k-1}. \tag{27}$$

When we define the state vector as the concatenation of the quaternion and the angular velocity vectors, we can apply the EKF on the following non-linear system:

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{q}_k \\ \boldsymbol{w}_k \end{bmatrix} = \boldsymbol{f}(\boldsymbol{x}_{k-1}), \tag{28}$$

$$\boldsymbol{z}_k = \mathbf{G}\boldsymbol{x}_k, \tag{29}$$

$$\mathbf{G} = \left[\mathbf{I}_{4\times4}\ \mathbf{0}_{4\times3}\right], \tag{30}$$

with $\boldsymbol{f}$ being the state transition function (which implies Eq. (24), (25), (26) and (27)), $\boldsymbol{z}_k$ the measured quaternion vector of length 4, and $\mathbf{G}$ the measurement matrix.

For implementing the EKF, it is essential to use the Jacobian of $\boldsymbol{f}$ which can be obtained by calculating the partial derivatives with respect to the state variables. For conciseness, the process noise and the measurement noise for the EKF are also omitted in the equations, but were considered in the implementation with their covariance matrices. Please note that $\boldsymbol{x}$, $\boldsymbol{z}$ are independent to those in Sect. 3.1, where we only intend to follow the conventional notations of the KF. Also, an inclusion of the angular acceleration in the rotation model may improve the performance depending on the tracking scenario.

4 Results

In this section, we propose a large series of experiments with our own data but also against multiple state of the art methods through a template-based tracking benchmark [15].

For our experiments, we used a USB3 *Pointgrey Flea3* camera with a 6 mm lens acquiring images with a spatial resolution of 1328 × 1048 pixels at 25 frames per second. Our method has been implemented on a computer with a 2 GHz processor and 4 GB of RAM. The initialization of the first frame is done using a simple rectangle detection algorithm.

Our tests focused on multiple aspects: the accuracy of the pose estimation and the speed improvement offered by our approach. Moreover, a comparison against multiple deterministic techniques is also proposed.

4.1 Pose Estimation

One important advantage of our approach is the direct computation of the camera's pose included in the optimization process itself. To highlight the accuracy offered by our method, we developed a practical assessment process which consists of capturing both the target image and a checkerboard - to accurately compute a ground truth pose. Our experimental platform is depicted in Fig. 2. In such a configuration, the coordinate system of the checkerboard and the target image are different, so we propose to compare the displacement of the camera with respect to its first position. The n^{th} camera motion $\mathbf{M}_{o1}^{on} = [\mathbf{R}_{o1}^{on}|\mathbf{t}_{o1}^{on}]$ is nothing but the composition of two transformations. From the checkerboard we compute the ground truth transformation ${}^{GT}\mathbf{M}_{o1}^{on} = \mathbf{M}_{c}^{on}(\mathbf{M}_{c}^{o1})^{-1}$, while the estimated motion from the tracking is computed as follow: $\mathbf{M}_{o1}^{on} = \mathbf{M}_{t}^{on}(\mathbf{M}_{t}^{o1})^{-1}$. Thus, the translational discrepancy can be calculated with: $e_t = \sqrt{\sum(\mathbf{t}_{o1}^{on} - {}^{GT}\mathbf{t}_{o1}^{on})}$, and the rotational error is computed as follow: $e_r = acos(\frac{1}{2}(tr((\mathbf{R}_{o1}^{on})^{-1}\ {}^{GT}\mathbf{R}_{o1}^{on})-1))$. As shown in Fig. 3, we acquired two sequences - *Van Gogh* (1500 images) and *Güell* (1200 images). In both videos a smooth and slow motion of the plane is performed within a distance range between 30 cm and 0.9 m. The pose estimation remains very accurate when the plane is close enough to the camera, as it is the case for the first sequence where the maximum distance is only 45 cm away. Indeed, for the *Van Gogh* sequence the maximum error is under 6 mm. Nonetheless, in the second sequence the distance is larger. In such circumstances the target covers a very small portion of the image which leads to a higher error in the pose estimation. Although the rotation remains accurate, the translational error is increasing up to 30 mm (for a distance of 0.9 m) which represents a percentile error of 3.3 %. This error is low enough for many applications, such as augmented reality.

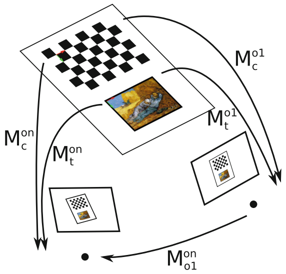

Fig. 2. Illustration of our pose estimation assessment

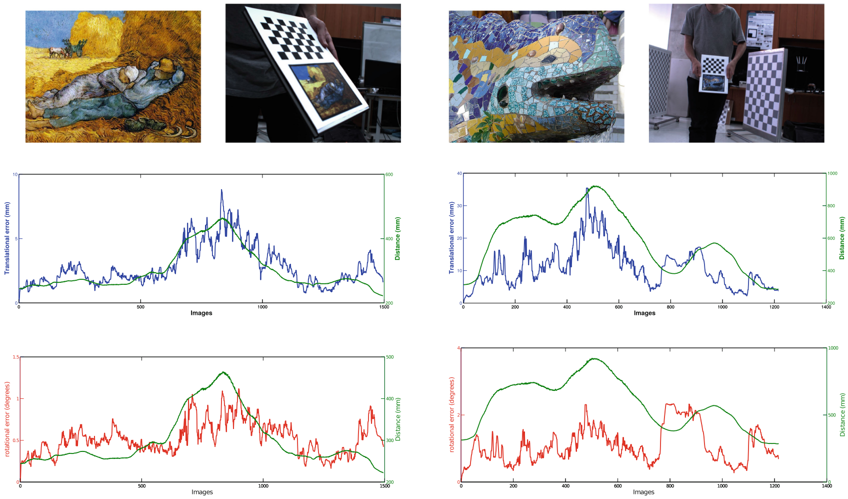

Fig. 3. Pose estimation results, (first row) Template and image sample from both sequences, (left column) Translational and rotational error of the *Van Gogh* sequence, (right column) Translational and rotational error of the *Güell* sequence.

4.2 Speed Evaluation

As claimed previously in this paper, the use of a predictive approach allows us to initialize the non-linear image alignment closer to the optimal minimum. Consequently, the number of needed iterations is reduced, which leads to a significant speed improvement of the algorithm. To emphasize this effect, we acquired a very challenging sequence called *Jung Seop* (which contains 1078 images).

This sequence consists of very fast motions at different distances. Figure 4(b) shows one representative tracking results obtained with our algorithm. On this figure, the green bounding box is the obtained result for the current image and the red bounding box is the previous position of the target which is usually utilized as an initialization in common approaches. Finally, the blue bounding box is the predicted pose from the EKF that we use to initialize our method. It is qualitatively clear that the prediction is closer to the final solution. More results are available in Fig. 5. We tried our algorithm with and without the EKF, and it showed that when the EKF is activated the sequence is fully tracked. Without this prediction step, the tracking failed after only 250 images due to very fast motions. Thus, we compared the required number of iterations in both cases only for the first 250 images of the sequence. For this test, the multi-resolution step is not utilized. We fixed the maximum number of iterations to 100, while the stopping criteria -the absolute difference between two successive updates ϵ - was defined by $\epsilon < 10^{-4}$. The results are available in Fig. 6, it is obvious that our method drastically reduced the number of iterations. For most of the images only 2 iterations are performed with the EKF prediction. In fact, the mean number of iterations with the EKF is 2.8 iterations per image while it is increased to 7.1 iterations without it.

Furthermore, ensuring the initialization to be in the vicinity of convergence leads to a more robust tracking. For instance, in Fig. 5, the warped images are strongly affected by motion blur, but even under these difficult conditions our method can efficiently track the target.

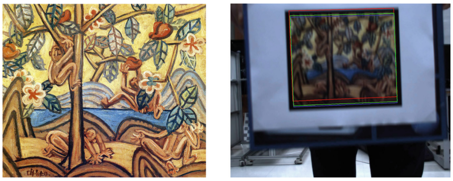

Fig. 4. *Jung Seop* sequence, (a) template image, (b) sample image of the tracking sequence

4.3 Benchmark Experiments

In order to compare our tracking approach against different state of the art methods, in this section we propose a full evaluation using the template-based tracking benchmark by Metaio GmbH [15]. This dataset consists of eight template images (see Fig. 7) with different characteristics: low, repetitive, normal and highly textured. Along with each template image, five video sequences are provided. The first sequence contains large angular motions and the second one includes challenging scale changes. In the third and fourth videos, fast far and fast close motions are performed. Finally, the last sequence is subject to strong illumination changes. The outputs of the tracker are compared with a ground

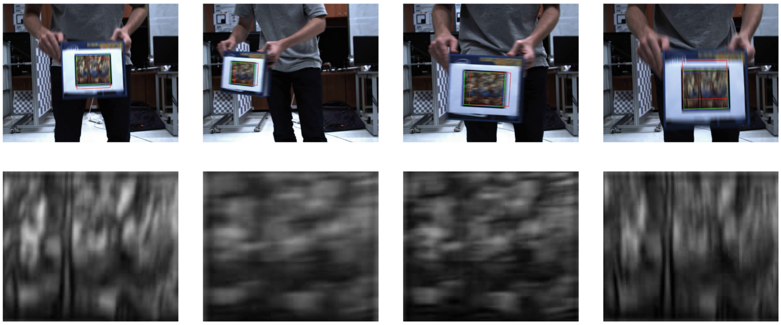

Fig. 5. Sample from the *Jung Seop* sequence, (first row) tracking results, (second row) corresponding warped images

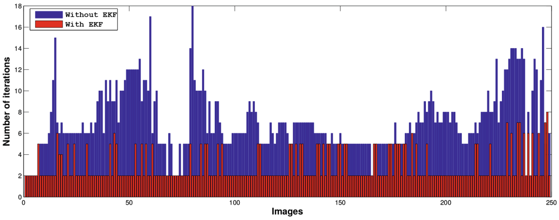

Fig. 6. Number of iterations per image with and without EKF

truth data; if the template position error is larger than 10 pixels, then the target is considered as lost. Consequently, the results are expressed in the percentage of successfully tracked frames.

We compare our method with three other approaches, the first being one of the most common template-based trackers, ESM [4]. The other methods are more recent and employ more efficient and robust objective functions, respectively, based on NCC [3] and MI [2].

For these sequences, our algorithm is configured with 3 multi-scale levels to increase the range of convergence, the EKF is activated and the stopping criterion is fixed at $\epsilon < 10^{-4}$ with a maximum of 20 iterations per scale level. Furthermore, the template image is downscaled to a size of 320×240 pixels to avoid oversampling.

Table 1 contains the results obtained from the algorithms where the highest scores are display in bold. The score difference below 5 % are not taken into account. According to this ranking, it is clear that our method significantly outperforms the compared deterministic approaches for almost every "fast" sequences, which can be directly attributed to the use of a EKF. Moreover, our algorithm also leads to better results under challenging motions such as large distance variations ("Range" sequences).

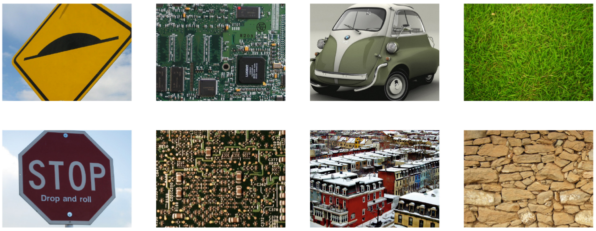

Fig. 7. Set of templates utilized in the benchmark [15], (from left to right column) low, repetitive, normal, high texturedness

Table 1. Ratio of successfully tracked images from the ESM [4], NCC [3], MI [2], and ours

ESM	Angle	Range	Fast Far	Fast Close	Illumination
Low	100.00%	92.33%	35.00%	21.58%	71.08%
	100.00%	64.17%	10.58%	26.83%	56.25%
Repetitive	61.92%	50.42%	22.50%	50.17%	34.50%
	2.92%	11.33%	6.83%	35.83%	11.33%
Normal	95.42%	77.75%	7.50%	67.08%	76.75%
	99.58%	99.00%	15.67%	86.75%	90.67%
High	0.00%	0.00%	0.00%	0.00%	0.00%
	100.00%	61.42%	22.83%	45.50%	79.67%

MI	Angle	Range	Fast Far	Fast Close	Illumination
Low	100.0%	94.1%	75.2%	56.5%	99.5%
	100.0%	98.1%	69.9%	43.7%	93.0%
Repetitive	76.9%	67.9%	22.8%	63.6%	100.0%
	91.3%	67.1%	10.4%	70.5%	96.2%
Normal	99.2%	99.3%	43.9%	86.7%	99.6%
	100.0%	100.0%	14.8%	84.5%	100.0%
High	47.1%	23.2%	7.2%	10.0%	50.6%
	100.0%	69.8%	20.8%	83.8%	100.0%

NCC	Angle	Range	Fast Far	Fast Close	Illumination
Low	99.7%	76.8%	52.7%	27.6%	100.0%
	100.0%	99.9%	21.6%	66.0%	**100.0%**
Repetitive	100.0%	57.7%	22.2%	68.2%	100.0%
	100.0%	81.3%	12.2%	53.6%	100.0%
Normal	100.0%	96.8%	58.2%	90.5%	100.0%
	99.9%	99.9%	20.1%	80.5%	100.0%
High	93.6%	**52.3%**	9.2%	14.0%	**98.9%**
	100.0%	51.5%	22.0%	75.0%	100.0%

ECC + EKF	Angle	Range	Fast Far	Fast Close	Illumination
Low	100.00%	**100.00%**	77.08%	**83.33%**	100.00%
	100.00%	100.00%	**100.00%**	**100.00%**	93.00%
Repetitive	100.00%	**100.00%**	**41.60%**	**85.40%**	100.00%
	100.00%	78.12%	**27.08%**	68.75%	100.00%
Normal	100.00%	100.00%	**93.75%**	89.58%	100.00%
	100.00%	100.00%	**54.16%**	**97.90%**	100.00%
High	**100.00%**	45.83%	**18.75%**	**41.60%**	79.16%
	100.0%	**79.16%**	**56.25%**	85.41%	100.00%

5 Conclusion

In this paper we proposed an efficient direct homography tracking algorithm able to deal with large motions. Our new formulation of the problem leads to two major improvements. Firstly, the pose can be accurately estimated in the tracking process itself, which reduces the number of DOF to 6. Secondly, this reformulation of the problem facilitates the addition of a predictive approach (the EKF) in the tracking process, while most of the state of the art method are purely deterministic. The predicted pose provides better initialization to the iterative image alignment process which allows the algorithm to cope with large motions, it also drastically improves the general robustness of the algorithm.

Many experiments have been proposed in this paper to highlight the advantages offered by our approach. Through these assessments it is clear that our algorithm outperforms state of the art methods for fast motions and provides a very accurate pose of the camera. Furthermore, the proposed method significantly reduces the number of iterations for the non-linear image alignment step.

Acknowledgement. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2010-0028680).

References

1. Evangelidis, G.D., Psarakis, E.Z.: Parametric image alignment using enhanced correlation coefficient maximization. IEEE Trans. Pattern Anal. Mach. Intell. **30**(10), 1858–1865 (2008)
2. Dame, A., Marchand, E.: Accurate real-time tracking using mutual information. In: ISMAR (2010)
3. Scandaroli, G.G., Meilland, M., Richa, R.: Improving NCC-based direct visual tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 442–455. Springer, Heidelberg (2012)
4. Benhimane, S., Malis, E.: Real-time image-based tracking of planes using efficient second-order minimization. In: IROS (2004)
5. Li, P., Chaumette, F., Tahri, O.: A shape tracking algorithm for visual servoing. In: ICRA (2005)
6. Mondragon, I., Campoy, P., Martinez, C., Olivares-Mendez, M.: 3d pose estimation based on planar object tracking for uavs control. In: ICRA (2010)
7. Kusuma Negara, G.P., Teck, F.W., Yiqun, L.: Hybrid feature and template based tracking for augmented reality application. In: Shan, S., Jawahar, C.V., Jawahar, C.V. (eds.) ACCV 2014 Workshops. LNCS, vol. 9010, pp. 381–395. Springer, Heidelberg (2015)
8. Park, Y., Lepetit, V., Woo, W.: Handling motion-blur in 3d tracking and rendering for augmented reality. IEEE Trans. Vis. Comput. Graphics **18**(9), 1449–1459 (2012)
9. Sorenson, H.W.: Kalman Filtering: Theory and Application. IEEE Press, New York (1985)
10. Broida, T., Chandrashekhar, S., Chellappa, R.: Recursive 3-d motion estimation from a monocular image sequence. IEEE Trans. Aerosp. Electron. Syst. **26**(4), 639–656 (1990)
11. Forster, C., Pizzoli, M., Scaramuzza, D.: Svo: fast semi-direct monocular visual odometry. In: ICRA (2014)
12. Silveira, G., Malis, E.: Real-time visual tracking under arbitrary illumination changes. In: CVPR (2007)
13. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Industr. Appl. Math. **11**(12), 431–441 (1963)
14. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Fluids Eng. **82**(1), 35–45 (1960)
15. Lieberknecht, S., Benhimane, S., Meier, P., Navab, N.: A dataset and evaluation methodology for template-based tracking algorithms. In: ISMAR (2009)