Tracking Feature Points in Uncalibrated Images with Radial Distortion

Miguel Lourenço and João Pedro Barreto

Institute for Systems and Robotics,
Dept. of Electrical and Computer Engineering,
University of Coimbra, Portugal
{miguel,jpbar}@isr.uc.pt

Abstract. The appearance of moving features in the field-of-view (FoV) of the camera may substantially change due to different camera poses. Typical solutions for tracking image points involve the assumption of an image motion model and the estimation of the motion parameters using image alignment techniques. While for conventional cameras this suffices, the radial distortion that arises in cameras with wide FoV lenses makes the standard motion models inaccurate. In this paper, we propose a set of motion models that implicitly encompass the distortion effect arising in this type of imaging devices. The proposed motion models are included in a standard image alignment framework for performing feature tracking in cameras presenting significant distortion. Consolidation experiments in repeatability and structure-from-motion scenarios show that the proposed RD-KLT trackers significantly improve the tracking performance in images presenting radial distortion, with minimal computational overhead when compared with a state-of-the-art KLT tracker.

1 Introduction

Tracking image keypoints across frames is useful in computer and robotic vision applications such as optical flow [1, 2], object tracking [3], and 3D reconstruction [4]. The interest in feature tracking dates back to [1, 2], where the authors propose the well known KLT tracker for computing optical flow between spatially and temporally close frames. The original KLT method assumes a translation model and iteratively estimates the displacement vector using image alignment techniques. Several improvements [5–8] have been proposed to the original method, specially aiming at reducing its computational complexity [5, 6] and improving tracking in wide-baseline situations [7, 8].

Wide field-of-view (FoV) cameras became increasingly popular due to their benefits in vision systems. Panoramic cameras proved to be highly advantageous in egomotion estimation [9, 10], and in surveillance systems due the thorough visual coverage of the environments [11]. However, the projection in cameras with wide angle lens presents strong radial distortion (RD) caused by the bending of the light rays when crossing the optics. The distortion increases with the distance to the center of distortion, and it is typically described by nonlinear terms that are function of the image radius.

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part IV, LNCS 7575, pp. 1–14, 2012.
© Springer-Verlag Berlin Heidelberg 2012

Image alignment techniques applied in a feature tracking context rely on the assumption of a motion model that determines the degree of deformation tolerated by the tracker. Several motion models have been used in the literature, ranging from a low complexity translation model [1, 2] to an affine motion model [5, 6, 8]. As discussed in [12] the performance of local feature tracking can be improved through the designed of specialized motion models. Unfortunately, the standard motion models do not compensate the RD effect arising in cameras equipped with unconventional optics.

Despite of these facts, the KLT tracker has been applied in the past to images with significant RD [13, 14]. While some simply ignore the effect of RD during registration [14], others correct the distortion in a pre-processing step before applying the KLT [13]. Although the later approach is quite straightforward, the distortion rectification requires the interpolation of the image signal, which is computationally expensive and unreliable since the synthetically corrected images contain artificially interpolated pixel intensities [15].

In this paper we focus on the problem of feature tracking in images presenting significant radial distortion. Our contributions are the following:

(i) We propose an extension of the affine motion model for describing the patches deformation that fuses feature motion with image distortion. It is proved that the proposed RD compensated motion model verifies the requirements to be used inside the efficient inverse compositional KLT framework [5, 6] whenever the calibration is known in advance. Unfortunately, the particular structure of this warp does not allow to calibrate the distortion during tracking, as we will discuss later;

(ii) To cope with this problem, we also propose an approximation to the ideal theoretical model that enables to robustly calibrate distortion during tracking. To the best of our knowledge this is the first work showing that is possible to estimate RD using solely low-level feature motion;

(iii) Extensive repeatability [16] and structure-from-motion experiments [15] show that the tracking performance can be significantly improved through a proper RD compensation, with a computational overhead of 15% when compared with a standard KLT algorithm.

The structure of this paper is as follows: Section 2 reviews the adopted camera model and the literature related with the KLT. Section 3 derives the RD compensated motion models and explains how to include them in the inverse compositional KLT. In section 4, the proposed RD-KLT trackers are evaluated in a representative set of repeatability [16] and structure-from-motion (SfM) experiments [15]. Finally, section 5 presents the conclusions of our work.

Notation: Matrices are represented by symbols in sans serif font, e.g. M, and image signals are denoted by symbols in typewriter font , e.g. $\mathtt{I}$. Vectors and vector functions are typically represented by bold symbols, and scalars are indicated by plain letters, e.g $\mathbf{x} = (x, y)^\mathsf{T}$ and $\mathbf{f}(\mathbf{x}) = (f_x(\mathbf{x}), f_y(\mathbf{x}))^\mathsf{T}$. $\mathbf{0}$ is specifically used to represent a null vector.

2 Background

In this section, we review the adopted camera model and the KLT framework using direct and inverse image alignment. We also summarize standard image motion models, and discuss the importance of the local template updates and pyramidal image representation for achieving reliable tracking.

2.1 The Division Model for Radial Distortion

We assume that the image distortion can be described using the 1st order division model with the amount of distortion being quantified by a single parameter ξ (typically $\xi < 0$). Let $\mathbf{x} = (x, y)^{\mathsf{T}}$ and $\mathbf{u} = (u, v)^{\mathsf{T}}$ be corresponding points in distorted and undistorted images expressed with respect to a reference frame with origin in the center of the image [17]. $\mathbf{f}$ is a vector function that maps points from the distorted image $\mathtt{I}$ to its undistorted counterpart $\mathtt{I}^u$:

$$\mathbf{u} = \mathbf{f}(\mathbf{x}) = (1 + \xi \mathbf{x}^{\mathsf{T}}\mathbf{x})^{-1}\mathbf{x}. \tag{1}$$

The function is bijective and the inverse mapping from $\mathtt{I}$ to $\mathtt{I}^u$ is given by[18]:

$$\mathbf{x} = \mathbf{f}^{-1}(\mathbf{u}) = 2(1 + \sqrt{1 - 4\xi \mathbf{u}^{\mathsf{T}}\mathbf{u}})^{-1}\mathbf{u}. \tag{2}$$

Given that the radius of $\mathbf{x}$ is $r = \sqrt{\mathbf{x}^{\mathsf{T}}\mathbf{x}}$, the corresponding undistorted radius is

$$r^u = (1 + \xi r^2)^{-1} r. \tag{3}$$

Henceforth, and in order to make the compression undergone by a particular image more intuitive, the amount of distortion will be quantified by

$$\%\,\mathrm{RD} = \frac{r^u_M - r_M}{r^u_M} \times 100 = -\xi r_M \times 100 \tag{4}$$

with r_M being the distance from the center to an image corner (maximum distorted radius) [15].

2.2 Kanade-Lucas-Tomasi Algorithm

Feature tracking between temporally adjacent images is typically formulated as a non-linear optimization problem whose cost function is the sum of the squared error between a template $\mathtt{T}$ and incoming images $\mathtt{I}$. The goal is to compute

$$\epsilon = \sum_{\mathbf{x} \in \mathcal{N}} \Big[\mathtt{I}(\mathbf{w}(\mathbf{x}; \mathbf{p})) - \mathtt{T}(\mathbf{x})\Big]^2, \tag{5}$$

where $\mathbf{p}$ denotes the components of the image warping function $\mathbf{w}$, and $\mathcal{N}$ denotes the integration region of a feature. Lucas and Kanade proposed to optimize

Eq. 5 by assuming that a current $\mathbf{p}$ motion vector is known and iteratively solve for $\delta\mathbf{p}$ increments on the warp parameters, with Eq. 5 begin approximated by

$$\epsilon = \sum_{\mathbf{x}\in\mathcal{N}} \Big[\mathtt{I}(\mathbf{w}(\mathbf{x};\mathbf{p}+\delta\mathbf{p}))-\mathtt{T}(\mathbf{x})\Big]^2 \approx \sum_{\mathbf{x}\in\mathcal{N}} \Big[\mathtt{I}(\mathbf{w}(\mathbf{x};\mathbf{p}))+\nabla\mathtt{I}\frac{\partial\mathbf{w}}{\partial\mathbf{p}}\delta\mathbf{p}-\mathtt{T}(\mathbf{x})\Big]^2. \quad (6)$$

Differentiating ϵ with respect to $\delta\mathbf{p}$, and after some algebraic manipulations, a closed-form solution for $\delta\mathbf{p}$ can be obtained:

$$\delta\mathbf{p} = \mathcal{H}^{-1}\sum_{\mathbf{x}\in\mathcal{N}}\Big[\nabla\mathtt{I}\frac{\partial\mathbf{w}(\mathbf{x};\mathbf{p})}{\partial\mathbf{p}}\Big]^{\mathsf{T}}\Big(\mathtt{T}(\mathbf{x})-\mathtt{I}(\mathbf{w}(\mathbf{x};\mathbf{p}))\Big), \quad (7)$$

with $\mathcal{H} = \sum_{\mathbf{x}\in\mathcal{N}}\Big[\nabla\mathtt{I}\frac{\partial\mathbf{w}(\mathbf{x};\mathbf{p})}{\partial\mathbf{p}}\Big]^{\mathsf{T}}\Big[\nabla\mathtt{I}\frac{\partial\mathbf{w}(\mathbf{x};\mathbf{p})}{\partial\mathbf{p}}\Big]$ being a 1st order approximation of the Hessian matrix, and the parameter vector being additively updated $\mathbf{p}^{i+1} \leftarrow \mathbf{p}^i + \delta\mathbf{p}$ at each iteration i. This method is also known as *forward additive* KLT [5, 6] and it requires to re-compute $\mathcal{H}$ at each iteration due its dependence with incoming image $\mathtt{I}$.

For efficiently solving Eq. 6, Baker and Matthews [5, 6] proposed an *inverse compositional alignment* method that starts by switching the roles of $\mathtt{T}$ and $\mathtt{I}$

$$\epsilon = \sum_{\mathbf{x}\in\mathcal{N}}\Big[\mathtt{I}(\mathbf{w}(\mathbf{x};\mathbf{p}))-\mathtt{T}(\mathbf{w}(\mathbf{x};\delta\mathbf{p}))\Big]^2 \approx \sum_{\mathbf{x}\in\mathcal{N}}\Big[\mathtt{I}(\mathbf{w}(\mathbf{x};\mathbf{p}))-\mathtt{T}(\mathbf{w}(\mathbf{x};\mathbf{0}))-\nabla\mathtt{T}\frac{\partial\mathbf{w}}{\partial\mathbf{p}}\delta\mathbf{p}\Big]^2. \quad (8)$$

The increments $\delta\mathbf{p}$ are then computed as:

$$\delta\mathbf{p} = \mathcal{H}^{-1}\sum_{\mathbf{x}\in\mathcal{N}}\Big[\nabla\mathtt{T}\frac{\partial\mathbf{w}(\mathbf{x};\mathbf{0})}{\partial\mathbf{p}}\Big]^{\mathsf{T}}\Big(\mathtt{I}(\mathbf{w}(\mathbf{x};\mathbf{p}))-\mathtt{T}(\mathbf{x})\Big), \quad (9)$$

with $\mathcal{H} = \sum_{\mathbf{x}\in\mathcal{N}}\Big[\nabla\mathtt{T}\frac{\partial\mathbf{w}(\mathbf{x};\mathbf{0})}{\partial\mathbf{p}}\Big]^{\mathsf{T}}\Big[\nabla\mathtt{T}\frac{\partial\mathbf{w}(\mathbf{x};\mathbf{0})}{\partial\mathbf{p}}\Big]$, and $\mathbf{w}(\mathbf{x};\mathbf{0})$ being the identity warp. $\mathcal{H}$ is computed using the template gradients and, therefore, it is constant during the registration procedure, leading to a significant computational improvement when compared with the forward additive KLT. Finally, the warp parameters are updated as follows:

$$\mathbf{w}(\mathbf{x};\mathbf{p}^{i+1}) \leftarrow \mathbf{w}(\mathbf{x};\mathbf{p}^i)\circ\mathbf{w}^{-1}(\mathbf{x};\delta\mathbf{p}). \quad (10)$$

Although the update rule of the inverse compositional alignment is computationally more costly than a simple additive rule, Baker and Matthews [5, 6] show that the overall computational complexity of the inverse formulation is significantly lower than that of the forward additive KLT.

The motion model $\mathbf{w}$ used for feature tracking determines the degree of image deformation tolerated during the registration process. The original contribution of Lucas and Kanade [1, 2] assumes that the neighborhood $\mathcal{N}$ around a feature

point $\mathbf{x}$ moves uniformly and, therefore, the authors model the image motion using a simple translation model. However, the deformation that it tolerates is not sufficient when the tracked image region is large, or the video sequence undergoes considerable changes in scale, rotation and viewpoint. In these situations, the affine motion model [5, 6, 8] is typically adopted

$$\mathbf{w}(\mathbf{x};\mathbf{p}) = (\mathsf{I} + \mathsf{A})\mathbf{x} + \mathbf{t}, \tag{11}$$

where the parameter vector is $\mathbf{p} = (a_1, .., a_4, t_x, t_y)^\mathsf{T}$, $\mathtt{I}$ is a 2×2 identity matrix, and $\mathsf{A} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$. In this paper, we propose an extension to the affine motion model that accounts for the RD effect arising in cameras equipped with wide FoV lenses.

For long-term feature tracking, the template update is a critical step to keep plausible tracks. An inherent problem to the template update step is the localization drift introduced whenever the template is updated [19]. High-order motion models tend to be more flexible in terms of the deformation tolerated during the registration process, with the templates being updated less frequently [19, 5, 6]. We carefully choose the frequency of the template update using the squared error of Eq. 5, as detailed in [8].

Despite of the warp complexity, the registration process may fail to converge when the initialization of the warp parameters $\mathbf{p}^0$ is not close enough to the current motion parameters, i.e. $\mathbf{p}^0$ is not in the convergence region $\mathcal{C}$ where the 1^{st} order approximation of Eq. 8 is valid [5, 6]. To attenuate this effect we adopt a pyramidal tracking framework [7], where several image resolutions are built by downsampling the image by factors of 2. A L-levels pyramidal tracking algorithm proceeds from the coarse to the finest pyramid level, with the coarsest feature position being given by $\mathbf{x}^L = 2^{-L}\mathbf{x}$. The registration proceed at each pyramid level, with the result begin propagated to next level as $\mathbf{x}^{L-1} = 2\,\mathbf{x}^L$ (for further details see [7]). Since the integration region $\mathcal{N}$ is kept constant across scales, the pyramidal framework greatly improves the probability of $\mathbf{p}^0$ belonging to $\mathcal{C}$, which by consequence increases the tracking success.

3 RD-KLT: Feature Tracking in Radial Distorted Images

In this section, we derive an extension to the affine motion model for cameras equipped with wide FoV lenses. It is proved that the derived RD model met the necessary requirements to be used in the inverse compositional KLT framework whenever the distortion calibration is known. As it will be discussed, this warping function does not allow to estimate the ξ during tracking due to its particular structure. Therefore, we also propose an approximation to the ideal theoretical model that enables to accurately estimate the distortion coefficient, at a negligible lost of tracking performance.

3.1 Mapping Composition for Deriving an RD Compensated Motion Model

Let's consider the standard situation where two undistorted images $\mathtt{I}^u$ and $\mathtt{I}^{u'}$ that are related by a generic motion function $\mathbf{w}$ such that $\mathtt{I}^u(\mathbf{u}) = \mathtt{I}^{u'}(\mathbf{w}(\mathbf{u};\mathbf{p}))$. We now consider that $\mathtt{I}^u$ and $\mathtt{I}^{u'}$ are the warping result of the original distorted images $\mathtt{I}$ and $\mathtt{I}'$. Using the distortion function of Eq. 1, we know that corresponding undistorted and distorted coordinates are related by $\mathbf{u} = \mathbf{f}(\mathbf{x})$, so we can re-write the mapping relation as $\mathtt{I}^u(\mathbf{u}) = \mathtt{I}^{u'}(\mathbf{w}(\mathbf{f}(\mathbf{x});\mathbf{p}))$. Since $\mathtt{I}^u(\mathbf{u}) = \mathtt{I}(\mathbf{x})$, with $\mathbf{x} = \mathbf{f}^{-1}(\mathbf{u})$, we can finally write directly the mapping relation between two distorted image signals as $\mathtt{I}(\mathbf{x}) = \mathtt{I}'(\mathbf{f}^{-1}(\mathbf{w}(\mathbf{f}(\mathbf{x});\mathbf{p})))$. Therefore, the RD compensated motion model that related the two distorted image signals can be expressed using the following function composition:

$$\mathbf{x}' = \mathbf{v}_\xi(\mathbf{x};\mathbf{p}) = \Big(\mathbf{f}^{-1} \circ \mathbf{w} \circ \mathbf{f}\Big)(\mathbf{x};\mathbf{p}). \tag{12}$$

3.2 cRD-KLT - Calibrated RD-KLT

In case the ξ coefficient is known in advance, the parameter vector $\mathbf{p}$ of $\mathbf{v}_\xi$ comprises the same parameters of the original motion of Eq. 11. The efficient inverse compositional KLT algorithm requires that the proposed set of warps form a *group* with respect to composition [5, 6]. The RD compensated motion model verifies the necessary *group* requirements:

(i) Identity - $\mathbf{v}_\xi(\mathbf{x};\mathbf{0}) = \mathbf{x}$
(ii) Invertibility - $\mathbf{v}_\xi(\mathbf{x};\mathbf{p})^{-1} = (\mathbf{f}^{-1} \circ \mathbf{v} \circ \mathbf{f})^{-1} = \mathbf{f}^{-1} \circ \mathbf{v}^{-1} \circ \mathbf{f}$
(iii) Composition - $\mathbf{v}_\xi(\mathbf{x};\mathbf{p}) \circ \mathbf{v}_\xi(\mathbf{x};\delta\mathbf{p}) = \mathbf{f}^{-1} \circ \mathbf{w}(\mathbf{x};\mathbf{p}) \circ \mathbf{w}(\mathbf{x};\delta\mathbf{p}) \circ \mathbf{f}$

It can be observed that the function composition to obtain the RD compensated model can be applied to any family of warps $\mathbf{w}$ that form group. By replacing our motion model $\mathbf{v}_\xi$ in the inverse composition KLT, it is straightforward to obtain the closed-form solution for $\delta\mathbf{p}$, which is given by:

$$\delta\mathbf{p} = \mathcal{H}_d^{-1} \sum_{\mathbf{x}\in\mathcal{N}} \left[\nabla\mathtt{T}\frac{\partial\mathbf{v}_\xi(\mathbf{x};\mathbf{0})}{\partial\mathbf{p}}\right]^\mathsf{T} \Big(\mathtt{I}(\mathbf{v}_\xi(\mathbf{x};\mathbf{p})) - \mathtt{T}(\mathbf{x})\Big) \tag{13}$$

with $\mathcal{H}_d = \sum_{\mathbf{x}\in\mathcal{N}} \left[\nabla\mathtt{T}\frac{\partial\mathbf{v}_\xi(\mathbf{x};\mathbf{0})}{\partial\delta\mathbf{p}}\right]^\mathsf{T} \left[\nabla\mathtt{T}\frac{\partial\mathbf{v}_\xi(\mathbf{x};\mathbf{0})}{\partial\mathbf{p}}\right]$, and the Jacobian $\frac{\partial\mathbf{v}_\xi(\mathbf{x};\mathbf{0})}{\partial\mathbf{p}}$ being evaluated at $\mathbf{p} = \mathbf{0}$. Finally, the motion parameters are updated at each iteration as follows:

$$\mathbf{v}_\xi(\mathbf{x};\mathbf{p}^{i+1}) \leftarrow \mathbf{v}_\xi(\mathbf{x};\mathbf{p}^i) \circ \mathbf{v}_\xi^{-1}(\mathbf{x};\delta\mathbf{p}) = \mathbf{f}^{-1} \circ \mathbf{w}(\mathbf{x};\mathbf{p}^i) \circ \mathbf{w}^{-1}(\mathbf{x};\delta\mathbf{p}) \circ \mathbf{f}. \tag{14}$$

3.3 Difficulties in Extending cRD-KLT to Handle Non-calibrated Images

The cRD-KLT considers a warping function $\mathbf{v}_\xi$ that compensates the radial distortion, applies the motion model, and then restores the non-linear image

deformation (see Fig. 1(a)). As it will be shown in the evaluation section, this approach is highly effective for performing image alignment of local patches in cameras with lens distortion, improving substantially the tracking accuracy and repeatability if compared with standard KLT. However, it has the drawback of requiring prior knowledge of the distortion parameter ξ, which implies a partial camera calibration. A strategy to overcome this limitation is to use the differential image alignment to estimate both the motion and the image distortion. This passes by extending the vector $\mathbf{p}$ of model parameters in order to consider ξ as a free variable in addition to the motion variables. In this case the warping function becomes $\mathbf{v}(\mathbf{x};\, \mathbf{q})$ with the difference with respect to $\mathbf{v}_\xi(\mathbf{x}, \mathbf{p})$ being only the vector $\mathbf{q} = (\mathbf{p}, \xi)$ of free parameters to be estimated.

Unfortunately, the model $\mathbf{v}(\mathbf{x};\, \mathbf{q})$ cannot be used for image registration using inverse compositional alignment. The problem is that any vector of parameters $\mathbf{q}$ of the form $\mathbf{q} = (\mathbf{0},\, \xi)$ is a null element that turns the warping function into the identity mapping

$$\mathbf{v}(\mathbf{x};\, (\mathbf{0},\, \xi)) = \mathbf{x},\, \forall_\xi.$$

This means that the Jacobian of $\mathbf{v}(\mathbf{x};\, \mathbf{q})$ evaluated for any $\mathbf{q}$ such that $\mathbf{p} = \mathbf{0}$ is singular and, consequently, $\mathcal{H}_d$ is non-invertible precluding the use of inverse compositional alignment. An alternative would be to use the forward additive framework, since the only requirement needed is the differentiability of the warp with respect to the motion parameters [5, 6]. Unfortunately, the computational complexity of this approach is significantly higher than that of the efficient inverse formulation. Instead of using the forward additive registration, the next section proposes to approximate the warp $\mathbf{v}(\mathbf{x};\, \mathbf{q})$ by assuming that the distortion is locally linear in a small neighborhood around the feature point.

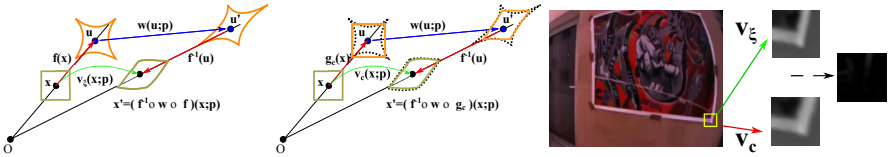
3.4 uRD-KLT - Uncalibrated RD-KLT

This section shows that it is possible to avoid the singular Jacobian issue by replacing the $\mathbf{v}(\mathbf{x}; \mathbf{q})$ by a suitable approximation of the desired composed warping. As it will be experimentally shown, this approximation has minimum impact in terms of error in image registration, enabling to use inverse compositional alignment to estimate both motion and distortion in an accurate and robust manner.

Let's assume that in a small neighborhood $\mathcal{N}$ around a feature $\mathbf{c}$ the distortion effect can be approximated by

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{g}_\mathbf{c}(\mathbf{x}) = (1 + \xi \mathbf{c}^\mathsf{T}\mathbf{c})^{-1}\mathbf{x}. \tag{15}$$

Remark that by replacing the radius of each point $\mathbf{x}$ by the radius of the central point $\mathbf{c}$ of the window $\mathcal{N}$ the non-linear function $\mathbf{f}$ becomes a projective transformation $\mathbf{g}_c(\mathbf{x})$ as shown in Fig. 1(b). This is a perfectly plausible approximation whenever the distance between the feature point $\mathbf{c}$ and the center of the image is substantially larger than the size of the neighborhood $\mathcal{N}$. In the situations where this is not verified, the effect of distortion is negligible, and the



(a) $\mathbf{v}_\xi$ (b) $\mathbf{v_c}$ (c) $\mathbf{v}_\xi - \mathbf{v_c}$

Fig. 1. Schematic difference between the (a) accurate and the (b) approximate RD compensated motion model. The black dashed lines in (b) represent the patches using the accurate RD model. (c) shows the difference between the accurate and the approximate models for a corner patch of an image with high distortion.

approximation does not introduce significant error. Replacing $\mathbf{f}$ by $\mathbf{g_c}$ in Eq. 12 yields the following approximation to the ideal theoretical model (see Fig.1(b)):

$$\mathbf{v_c}(\mathbf{x};\mathbf{q}) = \Big(\mathbf{f}^{-1} \circ \mathbf{w} \circ \mathbf{g_c}\Big)(\mathbf{x};\mathbf{q}). \tag{16}$$

In this case, the warp has single null element, and the Jacobian is not singular when evaluated in $\mathbf{q} = \mathbf{0}$, leading to an invertible $\mathcal{H}_d$. Remark that replacing $\mathbf{f}^{-1}$ by $\mathbf{g}_c^{-1}$ would again lead to a motion model with singular Jacobian and non-invertible $\mathcal{H}_d$.

Estimation of the Warp Parameters: The next step concerns the estimation of the increments $\delta\mathbf{q}$ of parameter vector $\mathbf{q}$. Due to the global nature of the RD, the distortion coefficient ξ must be simultaneously estimated for the N features being tracked, while keeping each the vector $\mathbf{p}$ specific for each feature. Recall that we want to compute the increment $\delta\mathbf{q}$ using the inverse compositional algorithm, through the following closed-form solution:

$$\delta\mathbf{q} = \mathcal{H}_d^{-1} \sum_{\mathcal{N}} \left[\nabla \mathtt{T} \frac{\partial \mathbf{v_c}(\mathbf{x};\mathbf{0})}{\partial \mathbf{p}}\right]^\mathsf{T} \Big(\mathtt{I}(\mathbf{v_c}(\mathbf{x};\mathbf{q})) - \mathtt{T}(\mathbf{x})\Big). \tag{17}$$

For each image feature, this equation can be re-written as

$$\mathsf{B}_{n\times n}\delta\mathbf{q}_{n\times 1} = \mathbf{e}_{n\times 1}, \tag{18}$$

where $\mathsf{B}_{n\times n} = \mathcal{H}_d = \big(\mathsf{H}_{n\times n-1}\ \mathbf{h}_{n\times 1}\big)$, and n is the number of parameters of $\mathbf{q}$. By performing a proper block-by-block stacking, the observation of all the N tracked features lead to the following system:

$$\underbrace{\begin{pmatrix} \mathsf{H}^1_{(n\times n-1)} & 0 & \dots & 0 & \mathbf{h}^1_{(n\times 1)} \\ 0 & \mathsf{H}^2_{(n\times n-1)} & & & \mathbf{h}^2_{(n\times 1)} \\ \vdots & & \ddots & & \vdots \\ 0 & \dots & 0 & \mathsf{H}^N_{(n\times n-1)} & \mathbf{h}^N_{(n\times 1)} \end{pmatrix}}_{\mathsf{B}_{nN\times(n-1)N+1}} \underbrace{\begin{pmatrix} \delta\mathbf{p}^1 \\ \delta\mathbf{p}^2 \\ \vdots \\ \delta\mathbf{p}^N \\ \delta\xi \end{pmatrix}}_{\delta\mathbf{q}^t_{(n-1)N+1\times 1}} = \underbrace{\begin{pmatrix} \mathbf{e}^1 \\ \mathbf{e}^2 \\ \vdots \\ \mathbf{e}^N \end{pmatrix}}_{\mathbf{e}^t_{nN\times 1}} \tag{19}$$

These systems of linear equations are typically solved through the computation of the pseudo-inverse $\delta\mathbf{q}^t = \mathsf{B}^\dagger\mathbf{e}^t = (\mathsf{B}^\mathsf{T}\mathsf{B})^{-1}\mathsf{B}^\mathsf{T}\mathbf{e}^t$. However, the explicit computation of the pseudo-inverse is computational expensive and subject to residual errors [20]. We solve the system of linear equations using the gaussian elimination method [20]. Since we have an over-constrained problem, we compute $\mathsf{B}^\mathsf{T}\mathsf{B}\delta\mathbf{q}^t = \mathsf{B}^\mathsf{T}\mathbf{e}^t$. Through Cholesky decomposition, we factorize $\mathsf{B}^\mathsf{T}\mathsf{B} = \mathsf{L}^\mathsf{T}\mathsf{L}$, with L being an upper triangular matrix. The updates $\delta\mathbf{q}^t$ are computed after solving an upper and lower triangular system, which are fast to compute [20].

Update of the Warp Parameters: The final step of the algorithm concerns the update the current parameters estimative. In theory [5, 6], the incremental warp $\mathbf{v_c}(\mathbf{x};\delta\mathbf{q})$ must be composed with the current warp estimative. We relax this composition requirement and use an approximate relation to update the warp parameters. We start from the relation given in [5, 6]

$$\mathbf{v_c}(\mathbf{x};\mathbf{q}^{i+1}) \leftarrow \mathbf{v_c}(\mathbf{x};\mathbf{q}^i) \circ \mathbf{v_c}^{-1}(\mathbf{x};\delta\mathbf{q}) \equiv \mathbf{v_c}(\mathbf{v_c}(\mathbf{x};-\delta\mathbf{q});\mathbf{q}^i). \tag{20}$$

Using this equation, we can formulate the parameters update as an additive step through the computation of a Jacobian matrix $\mathsf{J}_\mathbf{q}$ that maps the inverse compositional increment $\delta\mathbf{q}$ to its additive first-order equivalent $\mathsf{J}_\mathbf{q}\delta\mathbf{q}$ [5, 6], with the warp parameters being additively updated as $\mathbf{q}^{i+1} \leftarrow \mathbf{q}^i + \mathsf{J}_\mathbf{q}\delta\mathbf{q}$.

4 Experimental Validation

A tracking algorithm must be able to perform long-term feature tracking with high pixel accuracy [16]. Typically, the tracking performance is benchmarked through the evaluation of the tracking repeatability and the sub-pixel accuracy achieved during the image registration process [16]. This section compares the standard KLT algorithm against the proposed cRD-KLT and uRD-KLT trackers in sequences with different amounts of RD. All the trackers are directly used in the images with distortion, without ant type of rectification or pre-processing. We perform experiences in sequences of planar scenes, where it is possible to obtain ground truth to assess repeatability [16], and scenes with depth variation, where we evaluate the accuracy of Structure-from-Motion [15] . In addition, we describe an experience in self-calibration using the uRD-KLT tracker that can be helpful in practical surveillance scenarios. The three methods under evaluation were implemented using the affine motion model and a squared integration window $\mathcal{N}$ of 11×11 inside a pyramidal image registration with 4 resolution levels. Since our main goal is to perform feature (position) tracking rather than the template itself, we monitor the health of the template through the evaluation of the squared error of Eq. 5, with a new template being captured at the last feature position whenever required.

4.1 Repeatability Analysis in Planar Scenes

This experiment evaluates the reliability of the feature tracking algorithms using images of planar scenes. This means that every 2 images are related by an

homography that is used to verify the correctness and localization accuracy of the tracked features. For the computation of the ground truth homographies, we apply a robust estimation algorithm [21] that uses hundreds of correspondences obtained with sRD-SIFT, which provide precisely located features in radial distorted images [15]. The trackers are tested using four levels of distortion (0%, 10%, 25% and 45 %), with each level comprising 3 types of motion: slow translation, fast translation and generic camera motion.

We start by extracting 150 features using the Shi-Tomasi detection criteria [2], and track them along the 600 frames of each sequence. The reliability of the tracks are measured using the following metrics:

(i) *Repeatability* measures the ratio of correct points in the frame f using the ground truth homography H_1^f that provides the mapping from view 1 to f. The repeatability is measured as:

$$\mathcal{R} = \frac{\#(||\mathbf{x}_f - \mathsf{H}_1^f \mathbf{x}_1|| < \mathcal{D})}{\#(\mathsf{H}_1^f \mathbf{x}_1)}, \tag{21}$$

where $||\cdot||$ denotes the euclidean distance and $\mathcal{D} = 2$ pixels.

(ii) The *Sub-pixel accuracy* is measured for the points N that are reliably tracked. At frame f, we evaluate the RMS of the euclidean distance between consecutive feature positions as:

$$\mathcal{S}_{err} = \sqrt{\frac{\sum(||\mathbf{x}_f - \mathsf{H}_1^f \mathbf{x}_1||)^2}{N}}; \tag{22}$$

(iii) The *Photometric error* $\mathcal{A}_{err}$ is measured as the RMS of the squared error of Eq. 5 of the N tracked features.

We also include the computational time (FPS - frame per second) of the different methods for tracking the 150 features and the RD estimation for each level of distortion obtained using the uRD-KLT. The image sequences presenting distortion are calibrated using the Single Image Calibration (SIC) proposed in [22], which provides the ground truth for the distortion estimation.

Table 1 shows the repeatability results obtained in the planar image sequences. The conventional KLT tracker performs well in low distortion sequences, or when the motion between frames is smooth. In such cases, the distortion changes smoothly between two points locations, and the template update process enables to keep plausible tracks. However, when more complex motions, such as fast translation or affine camera motions are considered, the distortion changes more abruptly between two feature locations, precluding an effective performance of the registration process with direct consequences in the tracking results. As we increase the distortion and the complexity of the motion, the KLT starts loosing performance, which proves the importance of compensating distortion during tracking.

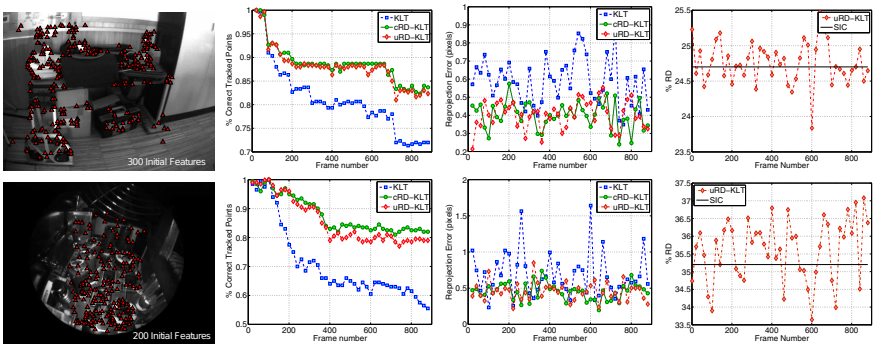
The compensation of distortion during registration, either by knowing RD calibration, or by performing it on-the-fly, brings improvements in all the evaluation parameters. The deformation tolerated by the RD compensated motion

Table 1. Performance evaluation in the planar scenes. The results are organized by type of motion (vertically) and corresponding amount of distortion (horizontally). The results presented are the RMS of the evaluation metric computed over the 600 frames. The distortion estimation and computational time are averaged over the 3 sequences with the same RD. The computational times were measured in a Intel Core i7-2600 CPU @3.4GHz.

				Slow Trans			**Fast Trans**			**Affine Motion**		
		%RD	FPS	$\mathcal{R}$	$\mathcal{S}_{err}$	$\mathcal{A}_{err}$	$\mathcal{R}$	$\mathcal{S}_{err}$	$\mathcal{A}_{err}$	$\mathcal{R}$	$\mathcal{S}_{err}$	$\mathcal{A}_{err}$
0%	**KLT**	—	6.11	0.98	0.21	0.014	0.95	0.27	0.021	0.90	0.35	0.032
	uRD-KLT	0.6±1.4	5.32	0.98	0.23	0.018	0.95	0.31	0.028	0.90	0.39	0.035
10%	**KLT**	—	6.09	0.98	0.38	0.038	0.92	0.58	0.055	0.90	0.59	0.045
	cRD-KLT	9.8	6.03	0.98	0.30	0.021	0.98	0.47	0.028	0.98	0.43	0.027
	uRD-KLT	9.4±0.48	5.28	0.98	0.32	0.021	0.98	0.47	0.028	0.98	0.43	0.027
25%	**KLT**	—	6.07	0.98	0.42	0.049	0.88	0.56	0.047	0.69	0.85	0.051
	cRD-KLT	24.7	6.02	0.99	0.33	0.026	0.98	0.43	0.026	0.90	0.55	0.027
	uRD-KLT	24.5±1.3	5.24	0.99	0.33	0.026	0.98	0.45	0.027	0.90	0.58	0.034
45%	**KLT**	—	5.95	0.87	0.81	0.051	0.76	1.15	0.065	0.64	1.27	0.076
	cRD-KLT	44.3	5.95	0.95	0.56	0.029	0.91	0.70	0.038	0.84	0.65	0.047
	uRD-KLT	44.2 ± 2.9	5.19	0.95	0.58	0.031	0.89	0.75	0.041	0.84	0.66	0.049

models allow to compensate the pernicious effects of distortion, which in practice is translated in accurate estimations of the feature motion parameters. This is visible in the lower appearance error and spatial accuracy achieved by the RD-KLT trackers. Since the registration is more accurate, the appearance error is lower, and the template update is less frequent, minimizing the inherent error in localization introduced by this process. It can also be observed that uRD-KLT performs slightly worse than the cRD-KLT algorithm in the sequences with high distortion and more complex motion. The differences in sub-pixel precision and photometric error are due to the use of the approximated RD motion model, which becomes slightly more imprecise as we increase distortion. Nevertheless, the difference is almost marginal without practical influence in the repeatability.

The 3 methods were implemented in Matlab/MEX files. The C-MEX files include operations that are transversal to the 3 methods, namely the interpolation routines, image gradient computation and image pyramid building. The computational time of the cRD-KLT ($\approx$ 1.11 milliseconds (ms)/feature) is slightly higher than the conventional KLT ($\approx$ 1.10 ms/feature). The small differences are explained by the different motion models used, which in our case is a non-linear mapping function that requires a little more computation. The uRD-KLT ($\approx$ 1.27 ms/feature) presents a computational overhead of $\approx$ 15%, which is a consequence of performing the RD estimation globally using Eq. 19. Nevertheless, it has the obvious advantage of not requiring distortion calibration for performing efficient feature tracking.



(a) Example image (b) Ratio of Inliers (c) Reproj. Error (d) RD estimation

Fig. 2. SfM experiments with a 25% distortion sequence and with a endoscopic sequence with 35% of RD. It can be observed that the RD-KLT tracker permit to long-term feature tracking (b) at a high precision accucary (c). (d) compares the distortion estimation form uRD-KLT with the explicit calibration results [22].

4.2 Structure-from-Motion (SfM)

Tracking features have been successfully applied to camera motion estimation and 3D scene reconstruction [21], with accurate point correspondence across frames being of key importance [21]. In this paper, the motion estimation is carried by a sequential SfM pipeline that uses as input the tracked points obtained by the 3 competing tracking methods. The objective is to recover the motion of two sparse sequences of 45 frames (sampled uniformly from sequences of 900 frames). The first sequence is obtained using a mini-lens that presents RD $\approx 25\%$, and the second sequence is captured using a boroscope with RD $\approx 35\%$, commonly used in medical endoscopy and industrial inspection.

The SfM pipeline iteratively adds new consecutive frames with a 5-point RANSAC initialization (using 2 views) [23], a scale factor adjustment (using 3 views) [21], and a final refinement with a sliding window bundle adjustment. Figure 2 shows that the motion estimation results. It can be observed that the RD-KLT trackers provide a lower re-projection error meaning that the extra parameter in the RD motion models permits a better convergence of the registration process in images presenting significant amounts of distortion. Finally, it can be seen in Fig. 2(d) that the distortion is robustly estimated, with the results being close to the ones obtained with the explicit calibration from [22].

4.3 RD Calibration for Surveillance Applications

Surveillance systems largely benefit with the usage of wide-angle lens that, due their wide FoV, enable a complete visual coverage of the environments [11]. In this final experiment, we show that using the uRD-KLT can be advantageous for estimating the distortion of a steady camera using the moving objects of

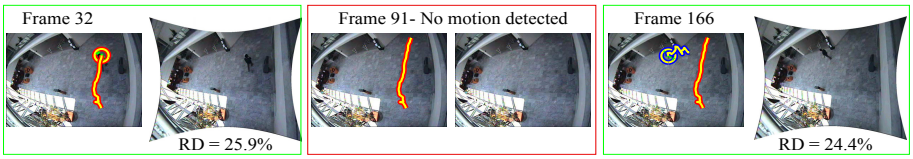


Fig. 3. Tracking experiment in a surveillance scenario from CAVIAR project. Distortion estimation is performed when significant motion is detected in the environment. Image inside the same bounding box concern the same instant of time. In each bounding box, the tracking results are shown on the left image, and the distortion estimation on the right image.

the scene. We test the algorithm using a sequence of the CAVIAR project[1], for which the RD calibration is unknown. We detect corner points at each frame sequence and initialize the uRD-KLT. If the points do not move in the next two frames, we re-initialize the tracker. The tracking results can be observed in Fig. 3. In each pair of bounded images, the original image (left image) shows the tracking results and the correspondent rectified image is shown on the right. In the middle block of images, the RD distortion estimated is negligible since no motion is detected and, therefore, the registration framework does not have any clues about how the local patches are deformed under the action of distortion.

5 Conclusions

This article presented for the first time an extension to the conventional KLT algorithm for point feature tracking in images with radial distortion. This was achieved by modifying the warping functions in order to account for both the motion and the non-linear image deformation arising in cameras with wide-angle lenses. Comparative experiments show that our RD-KLT tracker performs almost as well as the standard KLT tracker in sequences of correct perspective images, and achieves substantially better results in sequences with any amount of non-linear distortion. This is accomplished with minimum computational overhead. Such improvements in tracking are of strong importance for applications and domains that employ cameras equipped with mini-lens, fish-eye lenses, or boroscopes (e.g. robotics, medical applications, etc). In addition, we show for the first time that it is possible to accurately calibrate the image distortion while tracking low-level point features.

Acknowledgments. The authors acknowledge the Portuguese Science Foundation (FCT) that generously funded this work through grants PTDC/EIA-CCO/109120/2008 and SFRH/BD/63118/2009.

[1] Available at `http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/`

References

1. Lucas, B.D., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: DARPA Image Understanding Workshop, pp. 121–130 (1981)
2. Shi, J., Tomasi, C.: Good features to track. In: IEEE-CVPR, pp. 593–600 (1994)
3. Yilmaz, A., Javed, O., Shah, M.: Object Tracking: A survey. ACM Comput. Surv. 38 (2006)
4. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual Modeling with a Hand-Held Camera. IJCV 59, 207–232 (2004)
5. Baker, S., Matthews, I.: Equivalence and Efficiency of Image Alignment Algorithms. In: IEEE-CVPR, vol. 1, pp. 1090–1097 (2001)
6. Baker, S., Matthews, I.: Lucas-Kanade 20 Years On: A Unifying Framework. IJCV 56, 221–255 (2004)
7. Bouguet, J.Y.: Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm (2000)
8. Hwangbo, M., Kim, J.S., Kanade, T.: Gyro-aided feature tracking for a moving camera: fusion, auto-calibration and GPU implementation. IJRR 30, 1755–1774 (2011)
9. Gluckman, J., Nayar, S.: Egomotion and Omnidirectional Cameras. In: IEEE-ICCV (1998)
10. Baker, P., Fermuller, C., Aloimonos, Y., Pless, R.: A Spherical Eye from Multiple Cameras (Makes Better Models of the World). In: IEEE-CVPR (2001)
11. Caron, G., Eynard, D.: Multiple camera types simultaneous stereo calibration. In: IEEE-ICRA, pp. 2933–2938 (2011)
12. García Cifuentes, C., Sturzel, M., Jurie, F., Brostow, G.J.: Motion models that only work sometimes. In: BMVC (2012)
13. Koeser, K., Bartczak, B., Koch, R.: Robust GPU-assisted camera tracking using free-form surface models. Journal of Real-Time Image Processing 2, 133–147 (2007)
14. Behrens, A., Bommes, M., Stehle, T., Gross, S., Leonhardt, S., Aach, T.: Real-time image composition of bladder mosaics in fluorescence endoscopy. Computer Science - Research and Development 26, 51–64 (2011)
15. Lourenco, M., Barreto, J.P., Vasconcelos, F.: sRD-SIFT: Keypoint Detection and Matching in Images With Radial Distortion. In: IEEE-TRO (2012)
16. Gauglitz, S., Höllerer, T., Turk, M.: Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking. IJCV 94, 335–360 (2011)
17. Willson, R.G., Shafer, S.A.: What is the center of the image? J. Opt. Soc. Am. A 11, 2946–2955 (1994)
18. Barreto, J.P.: A Unifying Geometric Representation for Central Projection Systems. CVIU 103, 208–217 (2006)
19. Matthews, L., Ishikawa, T., Baker, S.: The Template Update Problem. IEEE-TPAMI 26, 810–815 (2004)
20. Davis, T.A.: Direct Methods for Sparse Linear Systems. Fundamentals of Algorithms, vol. 2. Society for Industrial and Applied Mathematics (2006)
21. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.: An Invitation to 3D Vision: From Images to Geometric Models. Springer (2003)
22. Barreto, J.P., Roquette, J., Sturm, P., Fonseca, F.: Automatic Camera Calibration Applied to Medical Endoscopy. In: BMVC (2009)
23. Nistér, D.: An Efficient Solution to the Five-Point Relative Pose Problem. IEEE-TPAMI 26 (2004)