J Math Imaging Vis (2016) 56:591–602
DOI 10.1007/s10851-016-0660-x

Multi-Camera Piecewise Planar Object Tracking with Mutual Information

Matthieu Fraissinet-Tachet[1] · **Michael Schmitt**[1] · **Zhuoman Wen**[2,3] · **Arjan Kuijper**[1]

Received: 26 October 2015 / Accepted: 27 April 2016 / Published online: 17 May 2016

Abstract Real-time and robust tracking of 3D objects based on a 3D model with multiple cameras is still an unsolved problem albeit relevant in many practical and industrial applications. Major problems are caused by appearance changes of the object. We present a template-based tracking algorithm for piecewise planar objects. It is robust against changes in the appearance of the object (occlusion, illumination variation, specularities). The version we propose supports multiple cameras. The method consists in minimizing the error between the observed images of the object and the warped images of the planes. We use the mutual information as registration function combined with an inverse composition approach for reducing the computational costs and get a near-real-time algorithm. We discuss different hypotheses that can be made for the optimization algorithm.

Keywords 3D object tracking · Model-based tracking · Template-based registration · Mutual information (MI) · Piecewise planar object

1 Introduction

Object tracking is a widely investigated area with many applications requiring different constraints on the algorithms that can be used. For offline methods, dedicated approaches can be used that are computationally expensive. For online methods, real-time (few milliseconds) or near-real-time (up to seconds) approaches are needed. As a consequence, a balance needs to be found between speed and accuracy or convergence of the algorithm. That is to say, the latter cannot be compromised too much, posing significant constraints on the methods that can be used. The aim of current research is therefore to design a precise and robust method for 3D model-based markerless object localization with multiple cameras and to discuss the validity of different approaches.

There is a wide field of possible applications: it can be used for visual servoing [5], in the context of augmented reality applications [20,28], surveillance [22] or pose estimation [11,12,14]. A typical industrial example (in for instance, car production) are robotic arms manipulating objects on a production line. They need to know the exact position and orientation of these objects in order to calculate their own trajectory in space. Multiple cameras may be used to supervise the task with the required precision. In this case, we know by advance the manipulated object and can therefore provide the algorithm with a 3D model. Under such conditions, a photo-realistic rendering of the object is not to be expected and the conditions of illumination or other appearance factors might change over time. Moreover, the algorithm will have to be robust enough to deal with occlusion. In the remainder of this paper we focus on these constraints.

The method we propose satisfies these constraints. We propose a novel approach for markerless object localization with multiple cameras. We made it near real time in combining the mutual information as registration function with an inverse composition approach. This creates a fast method that is robust against typical variations in appearance. In the remainder the main steps and results are presented. Full details can be found in the thesis [10].

✉ Matthieu Fraissinet-Tachet
matthieu.fraissinet-tachet@igd.fraunhofer.de

[1] Fraunhofer IGD, Fraunhoferstrae 5, 64283 Darmstadt, Germany

[2] Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China

[3] University of the Chinese Academy of Sciences, Beijing 100049, China

2 Related Work

Different approaches to visual tracking can be found in the literature. On the one hand we can use visual features—keypoints, lines or any other small geometrical shape—extracted from images in real time [21]. These shapes are local features and the algorithm relies then on their pairwise mapping. Here again we can distinguish two subcategories [25]: offline tracking and online tracking [29]. In the case of offline tracking, the key points from the current image are matched to precomputed keypoints from different views, also called keyframes, of the object. For online tracking, instead, the current keypoints are matched to the keypoints detected in the previous frame. The offline mode is robust, but does not take the estimation in last frame into account, while online tracking is quick and precise because it uses the last estimations. The same reasons lead to offline tracking being slow and not precise and online tracking prone to error accumulation. Vacchetti et al. merge the two approaches through local bundle adjustment to get an algorithm that is quicker and more robust [25]. Sometimes keypoints are falsely detected and perturb the pose estimation. One solution is to use robust statistics methods like RANSAC [9] to minimize the effect of these outliers. A 3D reconstruction of a piecewise planar scene can be done by detecting and matching feature points, so that it handles at the same time the construction and the tracking of the model [26]. Several difficulties arise from feature-based tracking [18]: outliers can lead to a less precise and stable pose estimation, surfaces that have an overall distinctive texture pattern but few distinctive local keypoints and also surfaces that show a significant curvature are harder to deal with. The KLT approach [15] uses local features for tracking images and also takes the spatial intensity information into account to direct and limit the search for the best match.

The alternative to using local features consists in making a template-based registration, in which the model is considered as a whole without paying attention to local features. It consists in warping the current image to a template image and maximizing the similarity between the two images based on the registration parameters. For 3D objects, Delabarre and Marchand [7] compare the current image with the model rendered at the estimated pose and then compute an update with a differential optimization algorithm. In the 2D case the object is a plane and the properties of homographies can be used to compute the warped image without the need to render the model anew [3,6]. It is even possible to use the inverse composition [1] implementation [6] so that some derivatives can be precomputed, making the process quicker. The planar 2D case can be generalized to 3D objects that can be represented as set of planes [2,7]. Indeed, for such an object, we can define the warping through a set of homographies depending on the pose of the object. As a matter of fact, a homography is defined by an underlying 3D transformation [16]. Delabarre and Marchand claim to use the inverse compositional approach applied to the 3D case without giving the details of the calculation [7]. We will focus on the use of the inverse compositional approach for piecewise planar objects. The problem being highly non-linear, we will consider the Levenberg–Marquardt Algorithm for the maximization of the similarity function, which is also used for similar problems [6,24]. Previous papers use one camera, we will explain how it is possible to integrate several cameras while keeping the low computational costs. This is, for instance, a good feature for a tracking system that is to be used for the supervision of a production line.

A similarity function is used to compare the set of intensity values of mapped points between the template image and the warped image. The most intuitive technique consists in using the sum of squared differences (SSD) to evaluate the pixel similarity but it behaves poorly when faced with illumination variation and occlusion [7]. The sum of conditional variance behaves better than SSD for global change of illumination but is not robust towards occlusion. Mutual Information (MI) [4] is a very robust similarity measure, much better than the two others, as it is robust against illumination variation, specularities and occlusions albeit at a high computational cost [6]. It is, e.g. widely used for medical image registration [13,19,27].

As Delabarre and Marchand [7] deal with the simultaneous tracking of planes to determine the position of the camera with a template-based registration method, we focus on the tracking of object that can be converted into a set of planes. The method for converting an object into a set of planes is beyond the scope of this paper, and therefore, we assume that we have an object made of planes. Tracking a piecewise planar object is easier because it is not necessary to render the object for each new position, so we can precompute a good part of the derivatives.

Our contribution is the integration of a multi-camera approach and the theoretical analysis of precisely one warp update in order to obtain a near-real-time multi-camera tracking approach.

3 Template-Based Registration

Our tracking algorithm is fully based on differential Template-Based Image registration, consisting of the optimization of an image registration function [7]. The goal is to find the position of an image template I^* on an image I. We note this position $\widehat{\boldsymbol{Z}}$. The problem with a similarity function f becomes

$$\widehat{\boldsymbol{Z}} = \arg\max_{\boldsymbol{Z}} f\left(I^*(\boldsymbol{x}), I(\omega_{\boldsymbol{Z}}(\boldsymbol{x}))\right), \tag{1}$$

where $\omega_{\boldsymbol{Z}}$ is the warp that associates a set of points $\boldsymbol{x}$ from the template I^* to the corresponding set of points $\omega_{\boldsymbol{Z}}(\boldsymbol{x})$ in I. The warp $\omega_{\boldsymbol{Z}}$ depends on $\boldsymbol{Z}$, which is the estimated relative position of I^* and I.

In our case, the position parameter $\boldsymbol{Z}$ will correspond to the pose of the 3D object and will be of dimension 6, as it is determined by its three rotation parameters and three translation parameters. Therefore, it is not reasonable to think about an exhaustive search of the state space for the estimation of the optimal pose. Instead we assume that we know an approximation $\boldsymbol{Z}_k$ of the current position and we look for a small displacement $\Delta\boldsymbol{Z}_k$ that leads to an improved similarity, i.e. a higher value of the similarity function.

There are two ways to formulate the update [1,7]: the direct compositional formulation and the inverse compositional formulation. In the latter case, the derivatives are computed on the reference image (the template) I^*, which is always the same. We can therefore precompute a good part of the derivatives and increase the performances. For this reason, we will focus our approach on the inverse compositional update. Thus, the warp update $\omega_{\Delta\boldsymbol{Z}}$ is computed as

$$\Delta\boldsymbol{Z}_k = \arg\max_{\Delta\boldsymbol{Z}} f\left(I^*\left(\omega_{\Delta\boldsymbol{Z}}(\boldsymbol{x})\right), I\left(\omega_{\boldsymbol{Z}_k}(\boldsymbol{x})\right)\right). \tag{2}$$

The warp update is $\omega_{\boldsymbol{Z}_{k+1}} = \omega_{\boldsymbol{Z}_k} \circ \omega^{-1}_{\Delta\boldsymbol{Z}_k}$. The similarity function f evaluates the similarity between the intensities of the set of pixels $\boldsymbol{x}$ from one image and the corresponding set of pixels $\omega(\boldsymbol{x})$ from the other image.

Mutual Information (MI) evaluates the quantity of information shared between two random variables, i.e. the distributions of the grey level intensities of two images. It is based on the histograms p_I, p_{I^*} and on the joint histogram p_{I^*I} of the two images:

$$\mathrm{MI}\big(I^*, I\big) = \sum_{r,s} p_{I^*I} \log \frac{p_{I^*I}}{p_{I^*}\, p_I}, \tag{3}$$

where r and s are histogram coordinates (bin indices). Histograms are computed using fuzzy binning with B-Splines of 3rd order [24,27] so that we can later compute the derivatives of the MI. What motivated our choice is that B-Splines are a good approximation of the Gaussians while the computation of their values and derivatives is cheaper.

In practice, we will try to use as few bins (intensity sampling) and as few points (image sampling) as possible in order to reduce the computational costs of the mutual information. This introduces a bias and a standard error that have been evaluated in [23]. The points are chosen randomly among a subset of pixels, for which the magnitude of the gradient is high enough. Indeed, the pose update will be noticeable as a variation of the MI value only if there is a significant intensity change for the pixels associated to the projected points (as the pose update changes the projection). As we want to get a better mapping, we look for a mapping that leads to a higher MI which is to say we have to optimize the MI with respect to the warping.

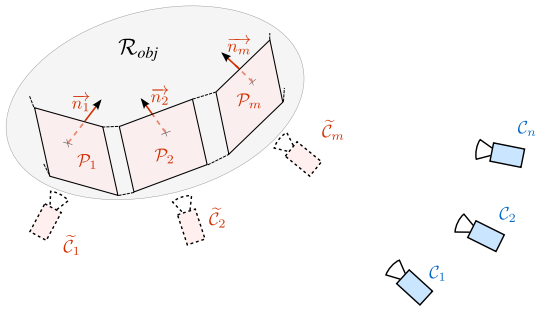


Fig. 1 Geometrical description of the model

In the next section, we explain how to construct the warping and its update based on the relative pose between the object and the cameras.

4 Description of the Geometric Model

In our system, represented by Fig. 1, we consider one object at the pose $T_{\mathcal{R}_{obj}}$ and a set of N real cameras $\mathcal{C}_n$ with $n \in \{1, \ldots, N\}$. A camera $\mathcal{C}_n$ is defined by its extrinsic and intrinsic parameters. The extrinsic parameter, written $T_{\mathcal{C}_n}$, transforms the world coordinates of a point into its coordinates in the local coordinate system of the camera $\mathcal{C}_n$. The intrinsic parameter, that we note $K_{\mathcal{C}_n}$, describes how a point in the camera coordinate system is projected on the camera image plane. The cameras are fixed and calibrated (extrinsic and intrinsic parameters are known).

The surface of the object is modelled by a set of M planes $\mathcal{P}_m$ with $m \in \{1, \ldots, M\}$. A plane $\mathcal{P}_m$ is defined by its normal $\overrightarrow{n_m}$ and by a point on that plane. As a plane corresponds to a local approximation of the surface of the object, we can decide that $\overrightarrow{n_m}$ is oriented in the direction of the inside of the object. Each plane has a texture which is rendered from a local virtual camera $\widetilde{\mathcal{C}}_m$, so that the plane coincides with the image plane of the virtual camera and the camera centre is in the negative space defined by the plane. The local camera is also defined by its intrinsic and extrinsic parameters, respectively, $T_{\widetilde{\mathcal{C}}_m}$ and $K_{\widetilde{\mathcal{C}}_m}$.

We need to define these local cameras because we will use homographies for warping a point from a plane $\mathcal{P}_m$ to its projection on the image plane of the camera $\mathcal{C}_n$. A homography predicts the warp between the images of two cameras looking at the same plane or, more generally, is a transformation from one projective plane to another. In our case, it is the homography between $\widetilde{\mathcal{C}}_m$ and $\mathcal{C}_n$ relative to the plane $\mathcal{P}_m$. Technically, the image plane of the camera $\widetilde{\mathcal{C}}_m$ does not need to coincide with the plane $\mathcal{P}_m$ (because of the properties

of homographies), but this is easier to understand it this way and there is no reason not to do so. Another specificity of these cameras is that they are fixed in the reference frame of the object, in other terms it does not move relatively to the object. This means that the camera $\widetilde{\mathcal{C}}_m$ always capture the same image of the plane $\mathcal{P}_m$ it is looking at, independently of the object displacement. $T_{\widetilde{\mathcal{C}}_m}$ is defined in the object coordinate system $\mathcal{R}_{obj}$. Local cameras are not physically present in the system, but they are required in our model because of the homographies.

4.1 Warping Update

Through the optimization process, we are looking for a small transformation update to apply to the pose of the object. We are looking for an update $T_{\Delta}^{(k)}$ such that

$$T_{\mathcal{R}_{obj}}^{(k+1)} = \left(T_{\Delta}^{(k)}\right)^{-1} T_{\mathcal{R}_{obj}}^{(k)}. \tag{4}$$

We use here the exponential map (see [2]) to parameterize $T_{\Delta}^{(k)}$ with $\theta_k \in \mathfrak{se}(3)$, so that $T_{\Delta}^{(k)} = T_{(\boldsymbol{\theta}_k)} = \exp(\boldsymbol{\theta}_k)$. We rewrite Eq. (4) in terms of $T_{\mathcal{R}_{obj}\to\mathcal{C}_n}$, the relative transformation between the object $\mathcal{R}_{obj}$ and the real camera $\mathcal{C}_n$:

$$T_{\mathcal{R}_{obj}\to\mathcal{C}_n}^{(k+1)} = T_{\mathcal{R}_{obj}\to\mathcal{C}_n}^{(k)} T_{(\boldsymbol{\theta}_k)}. \tag{5}$$

So we can think of $T_{(\boldsymbol{\theta}_k)}$ as an update of the relative transformation between $\mathcal{R}_{obj}$ and $\mathcal{C}_n$. Because of its definition in Eq. (4), the update is independent of the camera that is being considered.

We convert a point $\boldsymbol{X}_{\widetilde{\mathcal{C}}_m}$ belonging to $\mathcal{P}_m$ from the local camera $\widetilde{\mathcal{C}}_m$ coordinate system into the same point $\boldsymbol{X}_{\mathcal{C}_n}$ expressed in the coordinate system of the camera $\mathcal{C}_n$:

$$\boldsymbol{X}_{\mathcal{C}_n}^{(k+1)} = \widehat{T}_{\boldsymbol{\theta}_k} \boldsymbol{X}_{\widetilde{\mathcal{C}}_m}, \tag{6}$$

where $\widehat{T}_{\boldsymbol{\theta}_k} = T_{\mathcal{R}_{obj}\to\mathcal{C}_n}^{(k)} T_{(\boldsymbol{\theta}_k)} T_{\widetilde{\mathcal{C}}_m\to\mathcal{R}_{obj}}$.

The next step is to re-formulate it with homographies. We can do that because we are considering a map between two images (from a local camera $\widetilde{\mathcal{C}}_m$ and from a real camera $\mathcal{C}_n$) for pixels corresponding to the same plane $\mathcal{P}_m$. In terms of homographies, we have, respectively, before and after the update:

$$\begin{aligned} \boldsymbol{x}_{\mathcal{C}_n}^{(k)} &= K_{\mathcal{C}_n} H_{\widetilde{\mathcal{C}}_m} \left\{ T_{\mathcal{R}_{obj}\to\mathcal{C}_n}^{(k)} T_{\widetilde{\mathcal{C}}_m\to\mathcal{R}_{obj}} \right\} K_{\widetilde{\mathcal{C}}_m}^{-1} \boldsymbol{x}_{\widetilde{\mathcal{C}}_m} \\ \boldsymbol{x}_{\mathcal{C}_n}^{(k+1)} &= K_{\mathcal{C}_n} H_{\widetilde{\mathcal{C}}_m} \left\{ \widehat{T}_{\boldsymbol{\theta}_k} \right\} K_{\widetilde{\mathcal{C}}_m}^{-1} \boldsymbol{x}_{\widetilde{\mathcal{C}}_m}, \end{aligned} \tag{7}$$

where $H_{\widetilde{\mathcal{C}}_m}\{T\}$ is the homography looking at the plane $\mathcal{P}_m$ between the camera $\widetilde{\mathcal{C}}_m$ and $\widetilde{\mathcal{C}}_m$ when transformed by T, $K_{\mathcal{C}_n}$ is the intrinsic matrix of the camera $\mathcal{C}_n$, $K_{\widetilde{\mathcal{C}}_m}$ is the intrinsic matrix associated to camera $\widetilde{\mathcal{C}}_m$, $\boldsymbol{x}_{\widetilde{\mathcal{C}}_m}$ is the projection of a point $\boldsymbol{X}$ belonging to $\mathcal{P}_m$ on the image plane of the camera $\widetilde{\mathcal{C}}_m$ expressed in the 2D homogeneous coordinates, and $\boldsymbol{x}_{\mathcal{C}_n}^{(k)}$ is the projection of the same point $\boldsymbol{X}$ on the image plane of the camera $\mathcal{C}_n$ expressed in the 2D homogeneous coordinates at the iteration k.

Homographies require the use of 2D homogeneous coordinates, whereas we are interested in the pixel coordinates, which are euclidean. For this reason we consider warps, with the notation ω, instead of homographies.
We rewrite Eq. (7):

$$\boldsymbol{x}_{\mathcal{C}_n}^{(k+1)} = \omega_{\theta_k,m,n}\left(\boldsymbol{x}_{\widetilde{\mathcal{C}}_m}\right), \tag{8}$$

where

$$\omega_{\theta_k,m,n} = \omega_{\left[K_{\mathcal{C}_n} H_{\widetilde{\mathcal{C}}_m}\left\{\widehat{T}_{\boldsymbol{\theta}_k}\right\} K_{\widetilde{\mathcal{C}}_m}^{-1}\right]} \tag{9}$$

and $\omega_{[H]}$ decries the projection (euclidean coordinates) associated to the linear transformation H (in homogeneous coordinates). As we want to maximize the MI w.r.t. the exponential map parameter $\boldsymbol{\theta}$, the direct compositional formulation of the problem would be, based on Eq. (1):

$$\begin{aligned} \boldsymbol{\theta}^{\text{opt}} = \arg\max_{\boldsymbol{\theta}} MI\,(& I_{\widetilde{\mathcal{C}}_m}^{*}\left(\boldsymbol{x}_{\widetilde{\mathcal{C}}_m}\right), \\ & I_{\mathcal{C}_n}\left(\omega_{\boldsymbol{\theta},m,n}\left(\boldsymbol{x}_{\widetilde{\mathcal{C}}_m}\right)\right)\Big) \end{aligned} \tag{10}$$

where $\boldsymbol{\theta}^{\text{opt}}$ is the optimal parameter for defining the update.

From Eq. (10) we deduce, the inverse compositional update, which consists in computing the update (i.e. the derivatives) on the reference image $I_{\widetilde{\mathcal{C}}_m}^{*}$.
We first rewrite Eq. (10) as follows:

$$\begin{aligned} \boldsymbol{\theta}^{\text{opt}} = \arg\max_{\boldsymbol{\theta}} MI\,(& I_{\widetilde{\mathcal{C}}_m}^{*}\left(\omega_{\boldsymbol{\theta},m,n}^{-1}\left(\boldsymbol{x}'\right)\right), \\ & I_{\mathcal{C}_n}\left(\boldsymbol{x}'\right)\Big) \end{aligned} \tag{11}$$

so that we deduce the form that corresponds to the inverse compositional update:

$$\begin{aligned} \boldsymbol{\theta}^{\text{opt}} = \arg\max_{\boldsymbol{\theta}} MI\,(& I_{\widetilde{\mathcal{C}}_m}^{*}\left(\omega_{\boldsymbol{\theta},m,n}^{-1} \circ \omega_{\boldsymbol{0},m,n}\left(\boldsymbol{x}_{\widetilde{\mathcal{C}}_m}\right)\right), \\ & I_{\mathcal{C}_n}\left(\omega_{0,m,n}\left(\boldsymbol{x}_{\widetilde{\mathcal{C}}_m}\right)\right)\Big) \end{aligned} \tag{12}$$

This is indeed the inverse compositional form, because for $\boldsymbol{\theta} = 0$, we have

$$\omega_{\boldsymbol{\theta},m,n}^{-1} \circ \omega_{\boldsymbol{0},m,n} = \omega_{0,m,n}^{-1} \circ \omega_{\boldsymbol{0},m,n} = \text{Identity}.$$

If we use the notation $\delta\omega_{\boldsymbol{\theta},m,n} = \omega_{\boldsymbol{\theta},m,n}^{-1} \circ \omega_{\boldsymbol{0},m,n}$, we can simplify the expression Eq. (12):

$$\boldsymbol{\theta}^{\text{opt}} = \arg\max_{\boldsymbol{\theta}} MI\,(\, I^*_{\widetilde{\mathcal{C}}_m}\left(\delta\omega_{\boldsymbol{\theta},m,n}\left(\boldsymbol{x}_{\widetilde{\mathcal{C}}_m}\right)\right), \; I_{\mathcal{C}_n}\left(\omega_{0,m,n}\left(\boldsymbol{x}_{\widetilde{\mathcal{C}}_m}\right)\right)\Big) \tag{13}$$

For finding the optimal $\boldsymbol{\theta}$ we will use a differential optimization method and this requires the differentiation of the *MI* from equation Eq. (13).

The computation of the derivative will not be detailed here, but we can note that the computation is made easier with the fact that the function $\omega : H \longmapsto \omega_{[H]}$ for $H \in \mathbb{SL}$ is a homomorphism. It means $\omega_{[H_1 H_2]} = \omega_{[H_1]} \circ \omega_{[H_2]}$ for all $H_1, H_2 \in \mathbb{SL}$. From this, we can simplify the expression of the warp that is to be derived:

$$\begin{aligned}
&\omega^{-1}_{\boldsymbol{\theta},m,n} \circ \omega_{\mathbf{0},m,n} \\
&\quad = \omega^{-1}_{\left[K_{\mathcal{C}_n} H_{\widetilde{\mathcal{C}}_m}\{\widehat{T}_{\boldsymbol{\theta}}\} K^{-1}_{\widetilde{\mathcal{C}}_m}\right]} \circ \omega_{\left[K_{\mathcal{C}_n} H_{\widetilde{\mathcal{C}}_m}\{\widehat{T}_0\} K^{-1}_{\widetilde{\mathcal{C}}_m}\right]} \\
&\quad = \omega_{\left[\left(K_{\mathcal{C}_n} H_{\widetilde{\mathcal{C}}_m}\{\widehat{T}_{\boldsymbol{\theta}}\} K^{-1}_{\widetilde{\mathcal{C}}_m}\right)^{-1}\right]} \circ \omega_{\left[K_{\mathcal{C}_n} H_{\widetilde{\mathcal{C}}_m}\{\widehat{T}_0\} K^{-1}_{\widetilde{\mathcal{C}}_m}\right]} \\
&\quad = \omega_{\left[\left(K_{\mathcal{C}_n} H_{\widetilde{\mathcal{C}}_m}\{\widehat{T}_{\boldsymbol{\theta}}\} K^{-1}_{\widetilde{\mathcal{C}}_m}\right)^{-1} K_{\mathcal{C}_n} H_{\widetilde{\mathcal{C}}_m}\{\widehat{T}_0\} K^{-1}_{\widetilde{\mathcal{C}}_m}\right]} \\
&\quad = \omega_{\left[K_{\widetilde{\mathcal{C}}_m} H^{-1}_{\widetilde{\mathcal{C}}_m}\{\widehat{T}_{\boldsymbol{\theta}}\} H_{\widetilde{\mathcal{C}}_m}\{\widehat{T}_0\} K^{-1}_{\widetilde{\mathcal{C}}_m}\right]}.
\end{aligned} \tag{14}$$

4.2 Multiplane and Multi-camera Approach

The MI is used to evaluate the quality of the warp between two images. In our case, we are mapping the pixels from the image of the reference camera $\widetilde{\mathcal{C}}_m$ to the pixels of the camera $\mathcal{C}_n$. There are therefore as many warping as there are pairs $(\widetilde{\mathcal{C}}_m, \mathcal{C}_n)$. We compute the MI of all warps simultaneously:

$$MI = MI\left(\bigcup_{m,n}\left\{\left(I^*_{\widetilde{\mathcal{C}}_m}, I_{\mathcal{C}_n}\right)\right\}\right). \tag{15}$$

This is technically equivalent to compute the joint histograms $p_{\widetilde{\mathcal{C}}_m,\mathcal{C}_n}$ for each pair $(\widetilde{\mathcal{C}}_m, \mathcal{C}_n)$ and sum them all together in a linear combination (taking into account the number of points in each partial histograms) into p and then compute the mutual information out of the new global joint histogram p in the usual way.

The pose update that we defined in the previous section is still valid for the optimization of the global mutual information because we designed it to be independent of the considered pair of plane and camera. We have therefore

$$\boldsymbol{\theta}^{\text{opt}} = \arg\max_{\boldsymbol{\theta}} MI\left(\bigcup_{m,n}\left\{\left(\, I^*_{\widetilde{\mathcal{C}}_m}\left(\delta\omega_{\boldsymbol{\theta},m,n}\right),\; I_{\mathcal{C}_n}\left(\omega_{0,m,n}\right)\right)\right\}\right). \tag{16}$$

4.3 Mutual Information and Joint Histogram Derivatives

We use the Levenberg–Marquardt algorithm for finding the maximum of the mutual information, i.e. the finding the best pose. This algorithm requires the computation of the gradient (first-order derivatives) of the similarity function and also an estimation of its Hessian (second-order derivatives): The first-order and second-order derivatives of the MI are required to compute the object pose update with the Levenberg–Marquardt algorithm.

MI Gradient:

$$\left.\frac{\partial \text{MI}}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=0} = \sum_{r,s}\left[\frac{\partial p}{\partial \boldsymbol{\theta}} \log\frac{p}{p_1}\right]\Bigg|_{\boldsymbol{\theta}=0}. \tag{17}$$

MI Hessian:

$$\begin{aligned}
\left.\frac{\partial^2 \text{MI}}{\partial \boldsymbol{\theta}^2}\right|_{\boldsymbol{\theta}=0} &= \sum_{r,s}\left[\frac{1}{p}\frac{\partial p}{\partial \boldsymbol{\theta}}^T \frac{\partial p}{\partial \boldsymbol{\theta}}\right]\Bigg|_{\boldsymbol{\theta}=0} \\
&\quad - \sum_{r}\left[\frac{1}{p_1}\frac{\partial p_1}{\partial \boldsymbol{\theta}}^T \frac{\partial p_1}{\partial \boldsymbol{\theta}}\right]\Bigg|_{\boldsymbol{\theta}=0} \\
&\quad + \sum_{r,s}\left[\frac{\partial^2 p}{\partial \boldsymbol{\theta}^2}\log\frac{p}{p_1}\right]\Bigg|_{\boldsymbol{\theta}=0}.
\end{aligned} \tag{18}$$

The expression of the mutual information Hessian that we get is different from the one given by the paper [7]:

$$\begin{aligned}
\left.\frac{\partial^2 \text{MI}}{\partial \boldsymbol{\theta}^2}\right|_{\boldsymbol{\theta}=0} &= \sum_{r,s}\left[\frac{\partial p}{\partial \boldsymbol{\theta}}^T \frac{\partial p}{\partial \boldsymbol{\theta}}\left(\frac{1}{p} - \frac{1}{p_1}\right)\right]\Bigg|_{\boldsymbol{\theta}=0} \\
&\quad + \sum_{r,s}\left[\frac{\partial^2 p}{\partial \boldsymbol{\theta}^2}\log\frac{p}{p_1}\right]\Bigg|_{\boldsymbol{\theta}=0}.
\end{aligned} \tag{19}$$

The last expression relies on results from [8] which uses the following generally wrong assumption:

$$\sum_s \frac{1}{p_1(r)}\frac{\partial p(r,s)}{\partial \theta_i}\frac{\partial p_1(r)}{\partial \theta_j} = \sum_s \frac{1}{p_1(r)}\frac{\partial p(r,s)}{\partial \theta_i}\frac{\partial p(r,s)}{\partial \theta_j} \tag{20}$$

they implicitly use the rule $p_1(r) = \sum_s p(r,s)$, but when correctly used, it leads to

$$\begin{aligned}
&\sum_s \frac{1}{p_1(r)}\frac{\partial p(r,s)}{\partial \theta_i}\frac{\partial p_1(r)}{\partial \theta_j} \\
&\quad = \sum_s \frac{1}{p_1(r)}\frac{\partial p(r,s)}{\partial \theta_i}\frac{\partial}{\partial \theta_j}\sum_{s'} p(r,s') \\
&\quad = \sum_s\sum_{s'} \frac{1}{p_1(r)}\frac{\partial p(r,s)}{\partial \theta_i}\frac{\partial p(r,s')}{\partial \theta_j}.
\end{aligned} \tag{21}$$

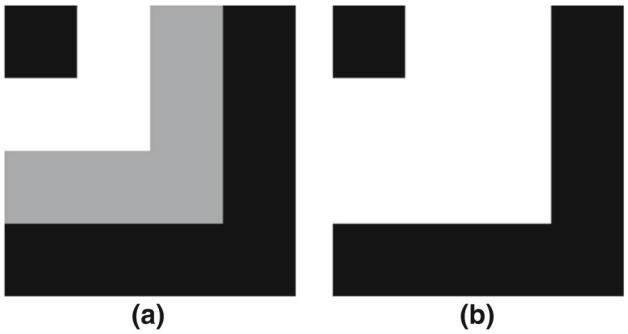


Fig. 2 Counter example for independency of histogram intensity bins at the convergence

There was therefore an index confusion, we cannot set $s' = s$.

Other papers [18,24] propose a *first-order approximation of the Hessian*, relying just on the first-order derivatives of the joint histogram:

$$\left(\widetilde{\frac{\partial^2 \mathrm{MI}}{\partial \boldsymbol{\theta}^2}}\right) := \sum_{r,s} \left[\frac{1}{p}\frac{\partial p}{\partial \boldsymbol{\theta}}^T \frac{\partial p}{\partial \boldsymbol{\theta}}\right]\Bigg|_{\boldsymbol{\theta}=0} - \sum_{r} \left[\frac{1}{p_1}\frac{\partial p_1}{\partial \boldsymbol{\theta}}^T \frac{\partial p_1}{\partial \boldsymbol{\theta}}\right]\Bigg|_{\boldsymbol{\theta}=0} \tag{22}$$

[24] justifies it assuming that in an ideal case and for two dependent images, $p(r,s) = p_1(r)p_2(s)$, which is to say that the population of the intensity ranges should be independent in the two images as p_2 does not depend on the update parameter. So the second-order part of the Hessian disappears:

$$\sum_{r,s}\left[\frac{\partial^2 p}{\partial \boldsymbol{\theta}^2}\log\frac{p}{p_1}\right] = \sum_{r,s}\left[\frac{\partial^2 p}{\partial \boldsymbol{\theta}^2}\log p_2\right] = \sum_{s}\left[\frac{\partial^2 p_2}{\partial \boldsymbol{\theta}^2}\log p_2\right] = 0. \tag{23}$$

This is true, for example, without fuzzy binning and if the images are identical (as the 2D histogram would be a diagonal matrix) but this is not true in general. Let us have a look at a counter example, based on the two images from Fig. 2. Figure 2b is the same image as Fig. 2a where the grey pixels have been replaced by white pixel. This is the kind of phenomenon that we have to deal with when performing a multimodal image registration. In this case, the optimal warping is the identity. Now if we build the joint histograms with 3 bins, we get

$$p = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0 & 0.3125 \\ 0 & 0 & 0.1875 \end{bmatrix},\ p1 = \begin{bmatrix} 0.5 \\ 0.3125 \\ 0.1875 \end{bmatrix},\ p2 = \begin{bmatrix} 0.5 & 0 & 0.5 \end{bmatrix}; \tag{24}$$

we conclude $p(3,3) \neq p_1(3)p_2(3)$ as $p(3,3) = 0.1875$ and $p_1(3)p_2(3) = 0.1875 \cdot 0.5 = 0.09375$. So the reason given by [24] is wrong.

Let us go further in the study of the first-order part of the Hessian. This is a symmetric matrix (which is an essential feature for a Hessian matrix) and we are going to show that it is likely to be positive semidefinite.
For all $v \in \mathbb{R}^6$:

$$v^t\left(\widetilde{\frac{\partial^2 \mathrm{MI}}{\partial \boldsymbol{\theta}^2}}\right)v = \sum_{r,s} v^t\left[\frac{1}{p}\frac{\partial p}{\partial \boldsymbol{\theta}}^T\frac{\partial p}{\partial \boldsymbol{\theta}}\right]v - \sum_{r} v^t\left[\frac{1}{p_1}\frac{\partial p_1}{\partial \boldsymbol{\theta}}^T\frac{\partial p_1}{\partial \boldsymbol{\theta}}\right]v = \sum_{r,s}\frac{1}{p}\left(\frac{\partial p}{\partial \boldsymbol{\theta}}\cdot v\right)^2 - \sum_{r}\frac{1}{p_1}\left(\frac{\partial p_1}{\partial \boldsymbol{\theta}}\cdot v\right)^2. \tag{25}$$

To prove that the Hessian is positive semidefinite, it is sufficient to prove that (25) is positive. We make use of Jensen's inequality, applied to the square function (which is convex):

$$\left(\frac{\partial p_1}{\partial \boldsymbol{\theta}}\cdot v\right)^2 = \left(\sum_s \frac{\partial p}{\partial \boldsymbol{\theta}}\cdot v\right)^2 = N_c^2\left(\sum_s \frac{1}{N_c}\frac{\partial p}{\partial \boldsymbol{\theta}}\cdot v\right)^2 \leq N_c^2\frac{1}{N_c}\sum_s\left(\frac{\partial p}{\partial \boldsymbol{\theta}}\cdot v\right)^2, \tag{26}$$

where N_c is the number of bins used for the computation of the histograms. This leads to

$$\left(\frac{\partial p_1}{\partial \boldsymbol{\theta}}\cdot v\right)^2 \leq N_c\sum_s\left(\frac{\partial p}{\partial \boldsymbol{\theta}}\cdot v\right)^2. \tag{27}$$

We can substitute the last inequality back in (25), so we get

$$v^t\left(\widetilde{\frac{\partial^2 \mathrm{MI}}{\partial \boldsymbol{\theta}^2}}\right)v \geq \sum_{r,s}\frac{1}{p}\left(\frac{\partial p}{\partial \boldsymbol{\theta}}\cdot v\right)^2 - \sum_{r,s}\frac{N_c}{p_1}\left(\frac{\partial p}{\partial \boldsymbol{\theta}}\cdot v\right)^2 = \sum_{r,s}\left(\frac{1}{p} - \frac{N_c}{p_1}\right)\left(\frac{\partial p}{\partial \boldsymbol{\theta}}\cdot v\right)^2. \tag{28}$$

As $\left(\frac{\partial p}{\partial \boldsymbol{\theta}} \cdot v\right)^2$ is always positive, we have to look at the sign of $\frac{1}{p} - \frac{N_c}{p_1}$. The problem is that as $p_1 = \sum_r p$, we conclude that $\frac{1}{p} - \frac{N_c}{p_1}$ can be positive and negative. However, we can show (briefly) that it is most of the times positive. Indeed $p = p(r, s)$ and $p_1 = p_1(r)$ can be seen as distributions for the random variables r and s. Therefore, if we consider that r and s are equally distributed, we can evaluate the expectation of $=\frac{1}{p} - \frac{N_c}{p_1}$:

$$\begin{aligned} E\left[\frac{1}{p} - \frac{N_c}{p_1}\right] &= \sum_{r,s} \frac{1}{N_c^2}\left(\frac{1}{p} - \frac{N_c}{p_1}\right) \\ &= \sum_r \frac{1}{N_c}\left(\sum_s \frac{1}{N_c}\frac{1}{p} - N_c \frac{1}{N_c}\frac{N_c}{p_1}\right). \end{aligned} \tag{29}$$

Here also, we apply the Jansen inequality to the function inverse, so that we have

$$\sum_s \frac{1}{N_c}\frac{1}{p} \geq \frac{1}{\sum_s \frac{1}{N_c} p} = \frac{1}{\frac{1}{N_c} p_1} = \frac{N_c}{p_1}. \tag{30}$$

So we finally have

$$\begin{aligned} E\left[\frac{1}{p} - \frac{N_c}{p_1}\right] &= \sum_{r,s} \frac{1}{N_c^2}\left(\frac{1}{p} - \frac{N_c}{p_1}\right) \\ &\geq \sum_r \frac{1}{N_c}\left(\frac{N_c}{p_1} - \frac{N_c}{p_1}\right) = 0. \end{aligned} \tag{31}$$

This means that the expectation of $\frac{1}{p} - \frac{N_c}{p_1}$ is positive. We cannot conclude directly for the positivity of $v^t \widetilde{\left(\frac{\partial^2 \mathrm{MI}}{\partial \boldsymbol{\theta}^2}\right)} v$ because we do not know the dependency between $\sum_{r,s}\left(\frac{1}{p} - \frac{N_c}{p_1}\right)$ and $\left(\frac{\partial p}{\partial \boldsymbol{\theta}} \cdot v\right)^2$. However, we can suppose that the two variables are independent, because the joint histogram is computed with the intensity values from the image and its derivatives with the image gradient. In an image there is in general no specific dependency between the intensity values and the gradient, so that we have

$$\begin{aligned} &\sum_{r,s}\left(\frac{1}{p} - \frac{N_c}{p_1}\right)\left(\frac{\partial p}{\partial \boldsymbol{\theta}} \cdot v\right)^2 \\ &\quad = E\left[\left(\frac{1}{p} - \frac{N_c}{p_1}\right)\left(\frac{\partial p}{\partial \boldsymbol{\theta}} \cdot v\right)^2\right] \\ &\quad = E\left[\frac{1}{p} - \frac{N_c}{p_1}\right] E\left[\left(\frac{\partial p}{\partial \boldsymbol{\theta}} \cdot v\right)^2\right] \geq 0. \end{aligned} \tag{32}$$

We can now conclude

$$\forall v \in \mathbb{R}^6, \quad v^t \widetilde{\left(\frac{\partial^2 \mathrm{MI}}{\partial \boldsymbol{\theta}^2}\right)} v \geq 0. \tag{33}$$

We have just proven that it is "likely" that the first approximation part of the Hessian is positive. The fact is that numerically it is even definite positive, so even if the proof is not rigorous, it gives an insight of the reason for that.

A priori the fact that the first-order approximation of the Hessian is positive is a really bad feature for a maximization problem, as the Hessian of a function evaluated on a local maximum is always negative. So first of all, we cannot call it an approximation anymore: the neglected part is big enough to make a positive matrix negative! We will call it the first-order part of the Hessian.

Then, how is it that [18,24] used this Hessian substitute successfully? The reason is that this first-order part of the Hessian still says something about the variations and more specifically about the curvature of the mutual information. If $\widetilde{\left(\frac{\partial^2 \mathrm{MI}}{\partial \boldsymbol{\theta}^2}\right)}$ is always semidefinite positive, then $-\widetilde{\left(\frac{\partial^2 \mathrm{MI}}{\partial \boldsymbol{\theta}^2}\right)}$ is semidefinite negative and this feature is perfect for the Newton part of the Levenberg–Marquardt algorithm. In fact, in the Newton process, the parameter generally converges to a local extremum, but if the Hessian used in the algorithm is negative, then the local extremum will be necessarily a local maximum. We just gave the explanation for the success that [18,24] encountered with the use of the first-order part of the Hessian. The effect of this substitution will be tested on the convergence and will be compared with the other Hessian evaluation methods in Sect. 6.1.

Another approach for the Hessian computation is to use its evaluation at convergence [7]. It is an interesting thing to be able to compute the Mutual Information derivatives at convergence. Indeed, the joint histogram computation is based on the pairs of intensities from a point on a reference image and its projection on the camera image. At the convergence, we can make an approximation and say that the intensity on the image camera equals the one from the reference image (as we said that mutual information is robust to multimodal images) and therefore use only the intensity values from the reference image to evaluate the derivatives of the mutual information on the convergence point.

This provides several benefits. If we use the evaluation of the Hessian at the convergence point, then we use the same Hessian value in all iteration steps and we can precompute it once for all. The other good point is that the Hessian is necessarily negative at the convergence, so it should improve the convergence radius, as explained for the first-order part of the Hessian.

We will try the same approach with the first-order part of the Hessian and test the algorithm substituting the Hessian with the opposite of its first-order part evaluated at the convergence, so that the precomputation is also cheaper.

The two last approaches will be also tested in Sect. 6.1.

4.4 Algorithm

We use Levenberg–Marquardt algorithm for the optimization of the MI as in [6]. Algorithm 1 describes the key steps of the optimization process and algorithm 2 is a subpart of it.

Algorithm 1 Optimization with global precomputations

Input
reference views $I_{\tilde{\mathcal{C}}_m}$, one for each plane $\mathcal{P}_m$
Initialize
for all planes **do**
select points used for MI estimation
compute the derivatives of the warped reference image
while new estimations with the same model **do**
Apply algorithm 2 to get the estimated pose

Algorithm 2 Levenberg Marquardt algorithm applied to the MI optimization

Input
precomputations (cf. algorithm 1)
N image $I_{\mathcal{C}_n}$, one per camera $\mathcal{C}_n$
initial pose guess $T^{(0)}_{\mathcal{R}_{obj}}$
Initialize
k=0
possibly: compute the MI Hessian at convergence using the visible planes
while Convergence criteria not met **do**
$k \leftarrow k+1$
while *true* **do**
Compute, at $\boldsymbol{\theta} = 0$: MI, $\frac{\partial MI}{\partial \boldsymbol{\theta}}$
possibly: compute, at $\boldsymbol{\theta} = 0$: $\frac{\partial^2 MI}{\partial \boldsymbol{\theta}^2}$
Compute $\boldsymbol{\theta}$ through the Newton update with Hessian regularization.
Try out the new parameters $MI(T_{(-\boldsymbol{\theta})}T^{(k-1)}_{\mathcal{R}_{obj}})$
if MI is increasing **then**
Accept parameter update $T^{(k)}_{\mathcal{R}_{obj}} \leftarrow T_{(-\boldsymbol{\theta})}T^{(k-1)}_{\mathcal{R}_{obj}}$
Decrease λ: $\lambda = \max(\frac{\lambda}{\lambda_{fact}}, \lambda_{min})$
Exit loop
else
Reject parameters and increase λ: $\lambda = \min(\lambda_{fact}\lambda, \lambda_{max})$
Check convergence criteria
Output
Estimated object pose $T^{(k)}_{\mathcal{R}_{obj}}$

5 Performance and Parameters

A natural criterion for evaluating the performance of an iterative algorithm such as Levenberg–Marquardt is its convergence radius. It consists in representing the convergence rate with respect to the intensity of the perturbation. In our experiments, we know the expected pose of the object so we decide that if the final pose estimation defines a mapping between the planes and the cameras with a *mean pixel error* under *1px*, then it has converged. Even if the algorithm does not converge, it is important that it becomes closer to the convergence point. Therefore, we also measure the error improvement: $\Delta e = \frac{e_{\text{end}} - e_{\text{start}}}{e_{\text{start}}}$. If a process converges, we also have to know how "quickly" it converges. As Levenberg–Marquardt is an iterative algorithm, we can first measure *how many iterations* are required till it converges. We also simply measure the *execution time* of the algorithm.

The Levenberg–Marquardt is an iterative algorithm and we need means of saying if the convergence point has been reached or not. If we know the optimal pose, we do not look at the distance between the estimated pose and the optimal pose (as a translation has a much smaller impact if the object is far away from the camera) but at what we perceive from this distance: the pixel error. We therefore monitor the mean pixel error (distance between the pixel projected with the estimated pose and those projected with the optimal pose) and declare that the algorithm is done when this error is small enough. If we do not know the real pose of the object there is no direct way to say if the estimated pose is close enough to the optimal pose and therefore we do not know a priori when to stop the iterations. We will look at two criteria for stopping the iterations in the next section.

6 Experiments

We test how the algorithm, in a multi-camera setup, responds to perturbations on the initial pose guess under different circumstances. In our experiments the object is made of two planes and we use two cameras to track the object (i.e. $M = 2$ and $N = 2$ according to the notations used in Sect. 4). An example is depicted Fig. 3.

The perturbation is defined by its parameter $\boldsymbol{\theta} \in \mathfrak{se}(3)$. In order to generate a perturbation, we apply a normal distribution on each coordinate of $\boldsymbol{\theta}$, centred in 0 and with different variance values. If we want to study the response of the algorithm to a perturbation which is a translation Tx along the

Fig. 3 Example of a generated experiment in a two camera setup for the tracking of an object made of two planes. The images depicted are the images taken from the two cameras

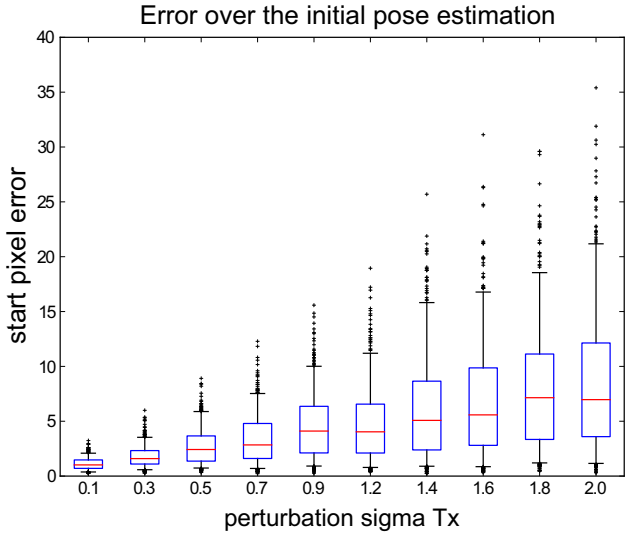


Fig. 4 Boxplot for the initial pixel error in function of the perturbation intensity (a translation along x-axis) for testing the estimation of the Hessian. The boxes delimit the 25 and 75% quartiles, while the whiskers delimit the 5 and 95% quantiles. The *red line* indicates the median and the points are the outliers (Color figure online)

x-axis then we will choose a very small variance for all other parameters than Tx and test the algorithm response to different perturbation magnitudes along Tx. We repeat each experiment 500 times with new randomly chosen parameters and measure the mean outcome, so that each point on a graph is the result of 500 trials.

The variance of the perturbation of the pose does not say much on how strong the perceived modifications of the scene are. For a same translation, the perceived modifications are much stronger if the camera is close to the object than if it is far away from it. Therefore, we will always display a graph representing the distribution of the mean start pixel error w.r.t. the magnitude of the perturbation (Fig. 4), so that we know which impact the perturbation has on the scene. For a given configuration we generate artificially the images of the cameras looking at the object.

We do not test the robustness of the algorithm against occlusion, illumination variation or specularities because it is a property that is inherited from the use of MI as similarity measure [7]. With the piecewise planar approach self-occlusion might happen, but parts of the planes that are occluded can be detected and filtered out before the computation of MI. In our experiment, the configuration is such that there is no self-occlusion, as only the invisible side of the plane can be occluded (the considered object is similar to concave).

The Levenberg–Marquardt algorithm requires to fix four λ (see algorithm 2). We set $\lambda_{\text{ref}} = 10^6$, $\lambda_{\text{fact}} = 10$, $\lambda_{\text{min}} = 10^{-6}$, and $\lambda_{\text{max}} = 10^8$ [18]. As explained in Sect. 3, the computation of the mutual information introduces two parameters: the number of histogram bins n_{bins} (intensity sampling) and the number of points n_{partial} per reference image or per plane (image sampling). Therefore, n_{partial} is also the number of points used for the evaluation of a partial joint histogram (see Sect. 4.2), so the total number of points used for the MI computation is $n_{\text{total}} = MNn_{\text{partial}}$. We set $n_{\text{bins}} = 14$ and $n_{\text{partial}} = 700$. It is also possible to choose $n_{\text{partial}} = 300$, for a quicker optimization and still get a satisfying convergence rate. Per plane, the points are randomly selected among the 30 percent of the points having the highest gradient (i.e. with gradient intensity higher than the 70th percentile). The more planes there are, the less points per plane are required: indeed, the decisive factor for the accuracy of the MI computation is the total number of points n_{total} in the system, since the planes are tracked simultaneously. The exact relationship between the accuracy of the MI and the total number of points (or in other terms the sampling points) is detailed in [23].

We can note that the complexity of the computation of MI is linear in the total number of points (as each point contribution is added to the joint histogram) and thus linear in the numbers of planes and cameras for a fixed number of points per plane. The complexity is also constant for varying numbers of planes and cameras as long as the number of points per plane is changed so that the total number of points stay constant (i.e. for a constant accuracy in the computation of the MI). For this reason, it is sufficient to carry out experiments with two planes and two cameras, as the system is scalable and the complexity is known.

6.1 Hessian Computation

There are different ways to compute an estimation of the Hessian. We implemented the following methods: PNC: first-order evaluated at each iteration step, PC: first-order evaluated at convergence, FNC: full Hessian evaluated at each iteration step and FC: full Hessian evaluated at convergence. These methods are detailed in Sect. 4.3.

Analysis of the convergence radius Figure 5a represents the convergence radius for the different Hessian computation modes. The first observation is that the convergence rate equals 1 for very small perturbations for all Hessian computation modes and decreases as the perturbation intensity increases. As expected, it is FNC that provides the best results: indeed the value of the MI Hessian is exact and completely coherent with the MI itself and therefore each update is good. FC, for which the full Hessian is estimated at convergence, ranks second. The graph shows that the first-order approximation for the Hessian computation has a clear impact on the convergence radius, as PNC and PC do not perform as well as FNC and FC. However, if the convergence

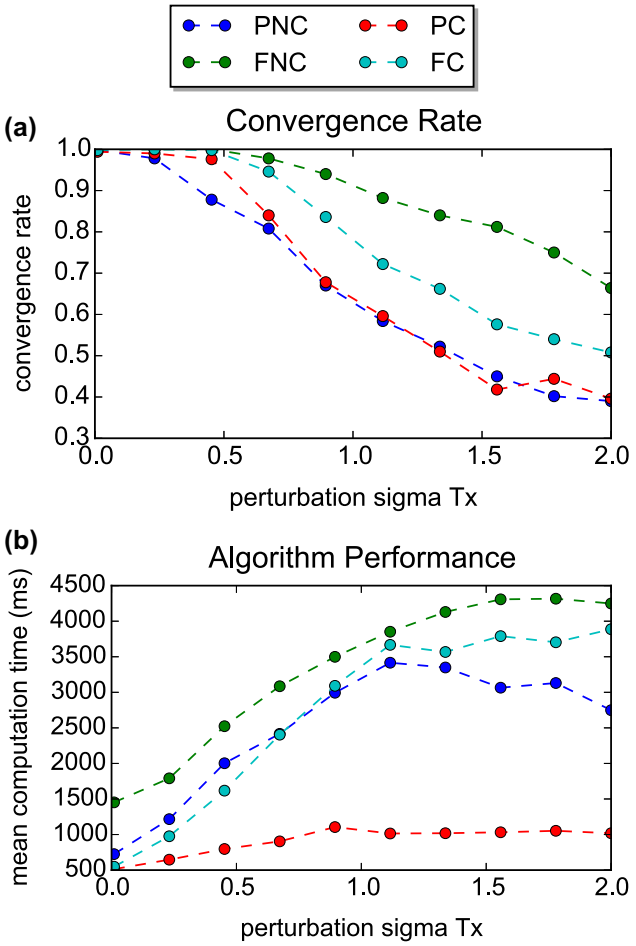


Fig. 5 **a** Convergence rate, **b** mean computation time for different Hessian computation modes

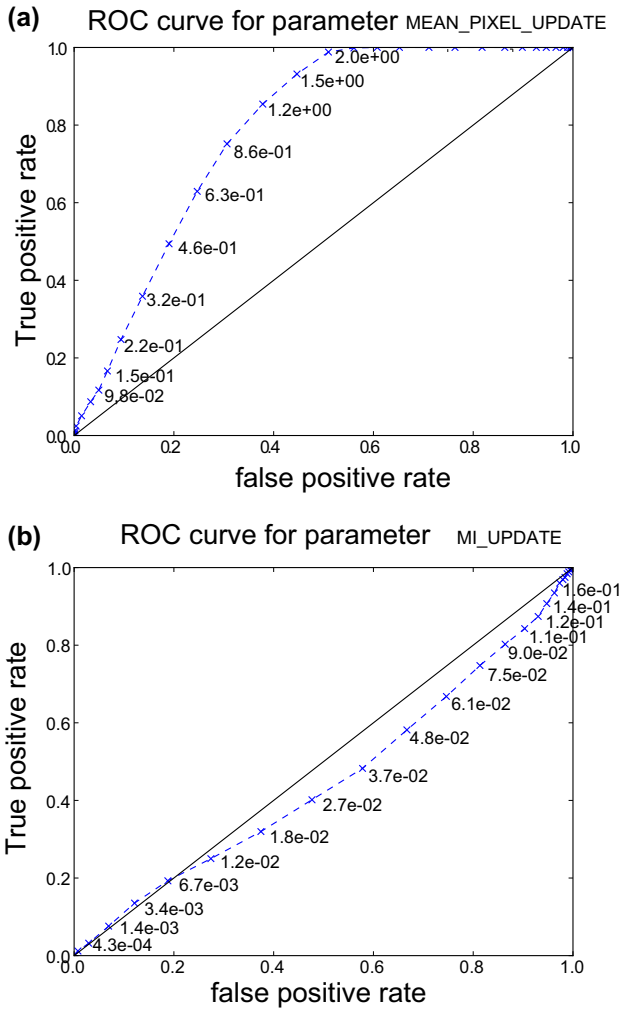


Fig. 6 ROC curves for mean pixel update (**a**) and MI update (**b**) criterion

is not so good for PNC and PC, it still converges and this corroborates the theoretical results from the Sect. 4.3.

Analysis of the computation time The computation time in each case is represented in Fig. 5b. Without surprise, FNC (without approximation) is the slowest method, whereas PC (two approximations) is the quickest, being nearly real time (it requires less than one second). This suggests that a trade-off between performance and convergence has to be made, depending on context where the algorithm is to be applied. The implementation of the algorithm is not optimized, and therefore it is useless to further comment Fig. 5. A better algorithm implementation is likely to reduce the computation time in all cases.

6.2 Stop Criteria

The Levenberg–Marquardt algorithm used for the optimization of the MI is an iterative algorithm. We need criteria in order to stop the optimization when we estimate that the solution is close enough to the convergence point.

The pose update is not relevant since a perturbation on the object pose has a smaller impact if the cameras are further away. In contrast, the measure of the mean pixel update accounts for the modification of the scene as perceived by the camera. Considering the update of MI per step also makes sense, as the MI update is likely to be smaller when close to the convergence point which is a local maximum of MI. For the two criteria (the MI update and the mean pixel update), and for different threshold values we look at the classification outcome and confront it with the true classification. The usual way for representing the performance of a binary classifier is to use ROC curve [17].

The ROC curve associated to the mean pixel update criterion (Fig. 6a) and MI update (Fig. 6b) shows that only the mean pixel update can be used as a criterion for evaluating the convergence.

7 Conclusion

We designed, implemented and tested a new approach for a precise pose estimation of a piecewise planar object based on an initial pose guess in near real time. It makes use of the MI as a similarity function so that it has the robustness required for industrial applications. The novelty is that the tracking is made in a multi-camera setup. We also precisely described the geometric structure of the scene and adapted it in terms of warps between planes and cameras scene using the properties of homographies.

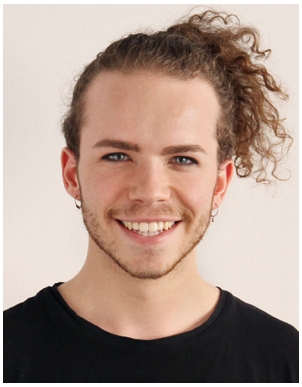
We designed the warp and its update so that it is possible to optimize the pose of the piecewise planar object exploiting the images and the poses of several cameras looking at the object. We also discussed the coherence of the method used in [7] for the Hessian estimation and identified a valid solution that gives the algorithm nearly real-time performances.

Acknowledgments This work has been developed in the Software Campus project CAD-VISION. CAD-VISION (Reference No: 01IS12053) is partly funded by the German ministry of education and research (BMBF) within the research programme ICT 2020.

References

1. Baker, S., Matthews, I.: Lucas-kanade 20 years on: a unifying framework. Int. J. Comput. Vis. **56**(3), 221–255 (2004). doi:10.1023/B:VISI.0000011205.11775.fd
2. Benhimane, S., Malis, E.: Integration of euclidean constraints in template based visual tracking of piecewise-planar scenes. In: Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pp. 1218–1223 (2006). doi:10.1109/IROS.2006.281859
3. Benhimane, S., Malis, E.: Homography-based 2d visual tracking and servoing. Int. J. Rob. Res. **26**(7), 661–676 (2007). doi:10.1177/0278364907080252
4. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, Hoboken (2012)
5. Dame, A., Marchand, E.: Mutual information-based visual servoing. IEEE Trans. Robot. **27**(5), 958–969 (2011). doi:10.1109/TRO.2011.2147090
6. Dame, A., Marchand, E.: Second-order optimization of mutual information for real-time image registration. IEEE Trans. Image Process. **21**(9), 4190–4203 (2012). doi:10.1109/TIP.2012.2199124
7. Delabarre, B., Marchand, E.: Camera localization using mutual information-based multiplane tracking. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pp. 1620–1625 (2013). doi:10.1109/IROS.2013.6696566
8. Dowson, N., Bowden, R.: A unifying framework for mutual information methods for use in non-linear optimisation. In: A. Leonardis, H. Bischof, A. Pinz (eds.) Computer Vision ECCV 2006, *Lecture Notes in Computer Science*, vol. 3951, pp. 365–378. Springer, Berlin Heidelberg (2006). doi:10.1007/11744023_29
9. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
10. Fraissinet-Tachet, M., Kuijper, A., Schmitt, M.: Mutual information-based piecewise planar object tracking. Master's thesis, Fraunhofer IGD, Technische Universitaet Darmstadt (2014)
11. Jiang, N., Cui, Z., Tan, P.: A global linear method for camera pose registration. In: IEEE International Conference on Computer Vision, ICCV 2013, Sydney, 1–8 Dec 2013, pp. 481–488 (2013). doi:10.1109/ICCV.2013.66
12. Kneip, L., Li, H.: Efficient computation of relative pose for multi-camera systems. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, 23–28 June 2014, pp. 446–453 (2014). doi:10.1109/CVPR.2014.64
13. Kuijper, A.: Mutual information aspects of scale space images. Pattern Recogn. **37**(12), 2361–2373 (2004). doi:10.1016/j.patcog.2004.04.014
14. Lee, G.H., Pollefeys, M., Fraundorfer, F.: Relative pose estimation for a multi-camera system with known vertical direction. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, 23–28 June 2014, pp. 540–547 (2014). doi:10.1109/CVPR.2014.76
15. Lucas, B.D., Kanade, T., et al.: An iterative image registration technique with an application to stereo vision. IJCAI **81**, 674–679 (1981)
16. Malis, E., Vargas, M.: Deeper understanding of the homography decomposition for vision-based control. Research Report RR-6303, INRIA (2007). http://hal.inria.fr/inria-00174036
17. Metz, C.E.: Basic principles of ROC analysis. Semin. Nucl. Med. **8**(4), 283–298 (1978)
18. Panin, G., Knoll, A.: Mutual information-based 3d object tracking. Int. J. Comput. Vision **78**(1), 107–118 (2008). doi:10.1007/s11263-007-0083-7
19. Pluim, J.P.W., Maintz, J., Viergever, M.: Mutual-information-based registration of medical images: a survey. IEEE Trans. Med. Imaging **22**(8), 986–1004 (2003). doi:10.1109/TMI.2003.815867
20. Prisacariu, V.A., Kähler, O., Murray, D.W., Reid, I.D.: Simultaneous 3d tracking and reconstruction on a mobile phone. In: IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2013, Adelaide, 1–4 Oct 2013, pp. 89–98 (2013). doi:10.1109/ISMAR.2013.6671768
21. Rios-Cabrera, R., Tuytelaars, T.: Discriminatively trained templates for 3d object detection: A real time scalable approach. In: IEEE International Conference on Computer Vision, ICCV 2013, Sydney, 1–8 Dec 2013, pp. 2048–2055 (2013). doi:10.1109/ICCV.2013.256
22. Rios-Cabrera, R., Tuytelaars, T., Gool, L.J.V.: Efficient multi-camera detection, tracking, and identification using a shared set of haar-features. In: The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, 20–25 June 2011, pp. 65–71 (2011). doi:10.1109/CVPR.2011.5995735
23. Roulston, M.S.: Estimating the errors on measured entropy and mutual information. Physica D: Nonlinear Phenomena **125**(34), 285–294 (1999). doi:10.1016/S0167-2789(98)00269-3 http://www.sciencedirect.com/science/article/pii/S0167278998002693
24. Thevenaz, P., Unser, M.: Optimization of mutual information for multiresolution image registration. IEEE Trans. Image Process. **9**(12), 2083–2099 (2000). doi:10.1109/83.887976
25. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3d tracking using online and offline information. IEEE Trans. Pattern Anal. Mach. Intell. **26**(10), 1385–1391 (2004)
26. Vigueras-Gomez, J.F., Sclaroff, S.: Fast vision-based scene modeling for augmented reality in unprepared man-made environments. J. Ambient Intell. Smart Environ. **5**(5), 525–537 (2013). http://dl.acm.org/citation.cfm?id=2594708.2594716
27. Viola, P., Wells, W.: Alignment by maximization of mutual information. In: Computer Vision, 1995. Proceedings., Fifth International Conference on, pp. 16–23 (1995). doi:10.1109/ICCV.1995.466930
28. Wagner, D., Reitmayr, G., Mulloni, A., Méndez, E., Diaz, S.: Mobile augmented reality - tracking, mapping and rendering. In: IEEE International Symposium on Mixed and Augmented Reality,

ISMAR 2014, Munich, 10–12 Sept 2014, p. 383 (2014). doi:10.1109/ISMAR.2014.6948500
29. Wu, Y., Lim, J., Yang, M.: Online object tracking: a benchmark. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, 23–28 June 2013, pp. 2411–2418 (2013). doi:10.1109/CVPR.2013.312



Matthieu Fraissinet-Tachet received his M.Sc. degree in Sciences of Engineering from Ecole Centrale de Lyon (France) and his M.Sc. degree in Computer Science from Technische Universitaet Darmstadt (Germany). His thesis "Mutual Information-Based Piecewise Planar Object Tracking" was awarded the best thesis award from the Visual Computing Cluster Darmstadt 2015. He currently works as a research associate at the Competence Center for Virtual and Augmented Reality at Fraunhofer IGD in Darmstadt. His research interests cover all aspects of mathematics-based methods for computer vision and augmented reality.

Michael Schmitt works at the Competence Center for Virtual and Augmented Reality at Fraunhofer IGD since he received his Master's Degree in Computer Science at TU Darmstadt in 2012. His research focuses on computer vision and its application in augmented reality.

Zhuoman Wen received her B.S. degree in electronic engineering from the Northeast Normal University in 2012. She is currently a Ph.D. student in the University of the Chinese Academy of Sciences. Her research interests include computer vision, artificial intelligence and digital image processing. E-mail: wenzhuoman@gmail.com

Arjan Kuijper holds a chair in "Mathematical and Applied Visual Computing" at TU Darmstadt and is a member of the management of Fraunhofer IGD, responsible for scientific dissemination. He received his M.Sc. degree in Applied Mathematics from Twente University and his Ph.D. from Utrecht University, both in the Netherlands. He was an assistant research professor at the IT University of Copenhagen, Denmark, and senior researcher at RICAM in Linz, Austria. He obtained his Habitation degree from TU Graz, Austria. He is the author of over 200 peer-reviewed publications, and serves as reviewer for many journals and conferences, and as program committee member and organizer of conferences. His research interests cover all aspects of mathematics-based methods for computer vision, graphics, imaging, pattern recognition, interaction and visualization.