Chapter 6
Homography Estimation Between Omnidirectional Cameras Without Point Correspondences

Robert Frohlich, Levente Tamás and Zoltan Kato

Abstract This chapter presents a novel approach for homography estimation between omnidirectional cameras. The solution is formulated in terms of a system of nonlinear equations. Each equation is generated by integrating a nonlinear function over corresponding image regions on the surface of the unit spheres representing the cameras. The method works without point correspondences or complex similarity metrics, using only a pair of corresponding planar regions extracted from the omnidirectional images. Relative pose of the cameras can be factorized from the estimated homography. The efficiency and robustness of the proposed method has been confirmed on both synthetic and real data.

6.1 Introduction

Homography estimation is essential in many applications including pose estimation (Sturm 2000), tracking (Mei et al. 2008; Caron et al. 2011), structure from motion (Makadia et al. 2007) as well as recent robotics applications with focus on navigation (Saurer et al. 2012), vision and perception (Molnár et al. 2014a, b). Efficient homography estimation methods exist for classical perspective cameras (Hartley and Zisserman 2003), but these methods are usually not reliable in case of omnidirectional sensors. The difficulty of homography estimation with omnidirec-

R. Frohlich · Z. Kato (✉)
Institute of Informatics, University of Szeged, Arpad ter 2, Szeged, Hungary
e-mail: kato@inf.u-szeged.hu

L. Tamás
Department of Automation, Technical University of Cluj-Napoca,
Dorobantilor st. 73, Cluj-Napoca, Romania
e-mail: Levente.Tamas@aut.utcluj.ro

R. Frohlich
e-mail: frohlich@inf.u-szeged.hu

Z. Kato
Department of Mathematics and Informatics, J. Selye University, Komarno, Slovakia

L. Buşoniu and L. Tamás (eds.), *Handling Uncertainty and Networked Structure in Robot Control*, Studies in Systems, Decision and Control 42,
DOI 10.1007/978-3-319-26327-4_6

tional cameras comes from the non-linear projection model yielding shape changes in the images that make the direct use of these methods nearly impossible.

Although non-conventional central cameras like catadioptric or dioptric (e.g. fish-eye) panoramic cameras have a more complex geometric model, their calibration also involves internal parameters and external pose. Recently, the geometric formulation of omnidirectional systems was extensively studied (Nayar 1997; Baker and Nayar 1999; Geyer and Daniilidis 2000; Mičušík and Pajdla 2004; Scaramuzza et al. 2006b; Puig and Guerrero 2013). The internal calibration of such cameras depends on these geometric models, which can be solved in a controlled environment, using special calibration patterns (Scaramuzza et al. 2006b; Kannala and Brandt 2006; Mei and Rives 2007; Puig and Guerrero 2013). When the camera is calibrated, which is typically the case in practical application, then image points can be lifted to the surface of a unit sphere providing a unified model independent of the inner non-linear projection of the camera. Unlike the projective case, homography is estimated using these spherical points (Mei et al. 2008; Caron et al. 2011). Of course, pose estimation must rely on the actual images taken in a real environment, hence we cannot rely on the availability of special calibration targets. A classical solution is to establish a set of point matches and then estimate homography based on these point pairs. For this purpose classical keypoint detectors, such as SIFT (Lowe 2004), are widely used (Makadia et al. 2007; Mei et al. 2008) for omnidirectional images.

Unfortunately, big variations in shape resolution and non-linear distortion challenges keypoint detectors as well as the extraction of invariant descriptors, which are key components of reliable point matching. For example, proper handling of scale-invariant feature extraction requires special considerations in case of omnidirectional sensors, yielding mathematically elegant but complex algorithms (Puig and Guerrero 2011). In Gutierrez et al. (2011) a new computation of descriptor patches was introduced for catadioptric omnidirectional cameras which also aims to reach rotation and scale invariance. In Makadia et al. (2007), a correspondence-less algorithm is proposed to recover relative camera motion. Although matching is avoided, SIFT features are still needed because camera motion is computed by integrating over all feature pairs that satisfy the epipolar constraint.

A number of works discuss the possibility of featureless image matching and recognition (most notably (Basri and Jacobs 1996)), but with limited success.

In this chapter, we propose a homography estimation algorithm which works directly on segmented planar patches. As a consequence, our method does not need extracted keypoints nor keypoint descriptors. In fact, we do not use any photometric information at all, hence our method can be used even for multimodal sensors. Since segmentation is required anyway in many real-life image analysis tasks, such regions may be available or straightforward to detect. Furthermore, segmentation is less affected by non-linear distortions when larger blobs are extracted. The main advantage of the proposed method is the use of regions instead of point correspondence and a generic problem formulation which allows to treat several types of cameras in the same framework. We reformulate homography estimation as a shape alignment problem, which can be efficiently solved in a similar way as in Domokos et al. (2012). The method has been quantitatively evaluated on a large synthetic dataset and proved

to be robust and efficient. We also show that the estimated homography can be used to recover relative pose of an omnidirectional camera pair both in the general case and, inspired by Saurer et al. (2012), under the *weak Manhattan world* assumption.

6.2 Planar Homography for Central Omnidirectional Cameras

A unified model for central omnidirectional cameras was proposed by Geyer and Daniilidis (2000), which represents central panoramic cameras as a projection onto the surface of a unit sphere $\mathcal{S}$ (see Fig. 6.1). According to Geyer and Daniilidis (2000), all central catadioptric cameras can be modeled by a unit sphere, such that the projection of 3D points can be performed in two steps: (1) the 3D point $\mathbf{X}$ is projected onto the unit sphere $\mathcal{S}$, obtaining the intersection $\mathbf{x}_{\mathcal{S}}$ of the sphere and the ray joining its center and $\mathbf{X}$ (see Fig. 6.1). (2) The spherical point $\mathbf{x}_{\mathcal{S}}$ is then mapped into the image plane $\mathcal{I}$ through the camera's internal projection function Φ yielding the image $\mathbf{x}$ of $\mathbf{X}$ in the omnidirectional camera. Thus a 3D point $\mathbf{X} \in \mathbb{R}^3$ in the camera coordinate system is projected onto $\mathcal{S}$ by central projection yielding the following relation between $\mathbf{X}$ and its image $\mathbf{x}$ in the omnidirectional camera:

$$\Phi(\mathbf{x}) = \mathbf{x}_{\mathcal{S}} = \frac{\mathbf{X}}{\|\mathbf{X}\|} \tag{6.1}$$

This formalism has been widely adopted and various models for the internal projection function Φ have been proposed by many researchers, e.g. Micusik (2004), Puig (2011), Scaramuzza (2006a) and Sturm (2011). From our point of view, Φ pro-

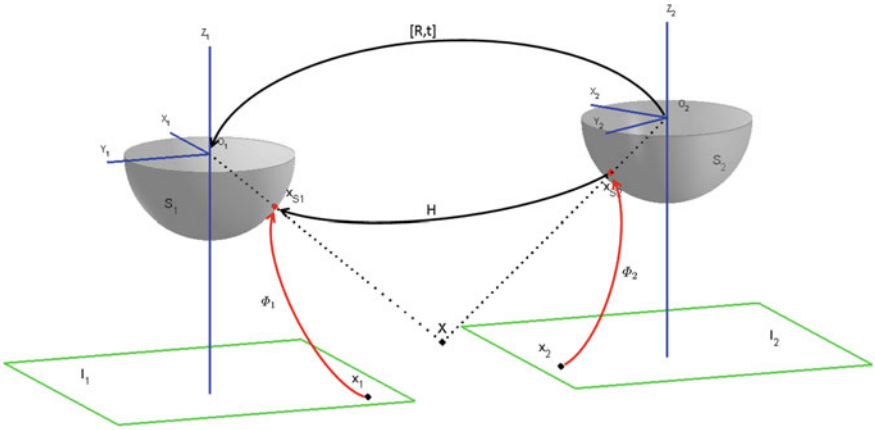

Fig. 6.1 Homography acting between omnidirectional cameras represented as unit spheres

vides an equivalent *spherical image* by backprojecting the omnidirectional image onto $\mathcal{S}$ and the planar homography $\mathbf{H}$ simply acts between these spherical images, as shown in Fig. 6.1.

Given a scene plane π, let us formulate the relation between its images $\mathcal{D}$ and $\mathcal{F}$ in two omnidirectional cameras represented by the unit spheres $\mathcal{S}_1$ and $\mathcal{S}_2$. The mapping of plane points $\mathbf{X}_\pi \in \pi$ to the camera spheres $\mathcal{S}_i, i = 1, 2$ is governed by (6.1), hence it is bijective (unless π is going through the camera center, in which case π is invisible). Assuming that the first camera coordinate system is the reference frame, let us denote the normal and distance of π to the origin by $\mathbf{n} = (n_1, n_2, n_3)^T$ and d, respectively. Furthermore, the relative pose of the second camera is composed of a rotation $\mathbf{R}$ and translation $\mathbf{t} = (t_1, t_2, t_3)^T$, acting between the cameras $\mathcal{S}_1$ and $\mathcal{S}_2$. Thus the image in the second camera of any 3D point $\mathbf{X}$ of the reference frame is

$$\mathbf{x}_{\mathcal{S}_2} = \frac{\mathbf{RX} + \mathbf{t}}{\|\mathbf{RX} + \mathbf{t}\|}$$

Because of the single viewpoint, planar homographies stay valid for omnidirectional cameras too (Mei et al. 2008). The standard planar homography $\mathbf{H}$ is composed up to a scale factor as

$$\mathbf{H} \propto \mathbf{R} + \frac{1}{d}\mathbf{t}\mathbf{n}^T \tag{6.2}$$

Basically, the homography transforms the rays as $\mathbf{x}_{\mathcal{S}_1} \propto \mathbf{H}\mathbf{x}_{\mathcal{S}_2}$, hence the transformation induced by the planar homography between the spherical points is also bijective. $\mathbf{H}$ is defined up to a scale factor, which can be fixed by choosing $h_{33} = 1$, i.e. dividing $\mathbf{H}$ with its last element, assuming it is non-zero. Note that $h_{33} = 0$ iff $\mathbf{H}(0, 0, 1)^T = (h_{13}, h_{23}, 0)^T$, i.e. iff the origin of the coordinate system in the first image is mapped to the ideal line in the second image. That happens only in extreme situations, e.g. when $Z_2 \perp Z$ and O_2 is on Z in Fig. 6.1, which is usually excluded by physical constraints in real applications. Thus the point $\mathbf{X}_\pi$ on the plane and its spherical images $\mathbf{x}_{\mathcal{S}_1}, \mathbf{x}_{\mathcal{S}_2}$ are related by

$$\mathbf{X}_\pi = \lambda_1 \mathbf{x}_{\mathcal{S}_1} = \lambda_2 \mathbf{H}\mathbf{x}_{\mathcal{S}_2} \Rightarrow \mathbf{x}_{\mathcal{S}_1} = \frac{\lambda_2}{\lambda_1}\mathbf{H}\mathbf{x}_{\mathcal{S}_2}$$

Hence $\mathbf{x}_{\mathcal{S}_1}$ and $\mathbf{H}\mathbf{x}_{\mathcal{S}_2}$ are on the same ray yielding

$$\mathbf{x}_{\mathcal{S}_1} = \frac{\mathbf{H}\mathbf{x}_{\mathcal{S}_2}}{\|\mathbf{H}\mathbf{x}_{\mathcal{S}_2}\|} \equiv \Psi(\mathbf{x}_{\mathcal{S}_2}) \tag{6.3}$$

6.3 Homography Estimation

Given a pair of omnidirectional cameras observing a planar surface patch, how to estimate the homography between its images, the spherical regions $\mathcal{D}_\mathcal{S} \in \mathcal{S}_1$ and $\mathcal{F}_\mathcal{S} \in \mathcal{S}_2$? First, let us formulate the relation between a pair of corresponding omni image points $\mathbf{x}_1$ and $\mathbf{x}_2$. The corresponding spherical points are obtained by applying the camera's inner projection functions Φ_1, Φ_2, which are then related by (6.3):

$$\Phi_1(\mathbf{x}_1) = \mathbf{x}_{\mathcal{S}_1} = \frac{\mathbf{H}\mathbf{x}_{\mathcal{S}_2}}{\|\mathbf{H}\mathbf{x}_{\mathcal{S}_2}\|} = \Psi(\Phi_2(\mathbf{x}_2)) \tag{6.4}$$

Any corresponding point pair $(\mathbf{x}_1, \mathbf{x}_2)$ satisfies the above equation. Thus a classical solution is to establish at least 4 such point correspondences $\{(\mathbf{x}_1^i, \mathbf{x}_2^i)\}_{i=1}^N$ by standard intensity-based point matching, and solve for $\mathbf{H}$. However, the inherent non-linear distortion of omnidirectional imaging challenges traditional keypoint detectors as well as the extraction of invariant descriptors, which are key components of reliable point matching. Therefore we are interested in a solution without finding point matches.

We will show that by identifying a single planar region in both omni images (denoted by $\mathcal{D}$ and $\mathcal{F}$, respectively), $\mathbf{H}$ can be estimated without any additional information. Since we do not have established point pairs, we cannot directly use (6.4). However, we can get rid of individual point matches by integrating both sides of (6.4) yielding a surface integral on $\mathcal{S}_1$ over the surface patches $\mathcal{D}_\mathcal{S} = \Phi_1(\mathcal{D})$ obtained by lifting the first omni image region $\mathcal{D}$ and $\mathcal{F}_\mathcal{S} = \Psi(\Phi_2(\mathcal{F}))$ obtained by lifting the second omni image region $\mathcal{F}$ and transforming it by $\Psi : \mathcal{S}_2 \to \mathcal{S}_1$. To get an explicit formula for these integrals, the surface patches $\mathcal{D}_\mathcal{S}$ and $\mathcal{F}_\mathcal{S}$ can be naturally parameterized via Φ_1 and $\Psi \circ \Phi_2$ over the planar regions $\mathcal{D} \subset \mathbb{R}^2$ and $\mathcal{F} \subset \mathbb{R}^2$:

$$\begin{aligned}
&\forall \mathbf{x}_{\mathcal{S}_1} \in \mathcal{D}_\mathcal{S} : \mathbf{x}_{\mathcal{S}_1} = \Phi_1(\mathbf{x}_1), \mathbf{x}_1 \in \mathcal{D} \\
&\forall \mathbf{z}_{\mathcal{S}_1} \in \mathcal{F}_\mathcal{S} : \mathbf{z}_{\mathcal{S}_1} = \Psi(\Phi_2(\mathbf{x}_2)), \mathbf{x}_2 \in \mathcal{F},
\end{aligned}$$

yielding the following integral equation:

$$\begin{aligned}
\iint_\mathcal{D} \Phi_1(\mathbf{x}_1) \left\| \frac{\partial \Phi_1}{\partial x_{11}} \times \frac{\partial \Phi_1}{\partial x_{12}} \right\| \mathrm{d}x_{11}\, \mathrm{d}x_{12} = \\
\iint_\mathcal{F} \Psi(\Phi_2(\mathbf{x}_2)) \left\| \frac{\partial (\Psi \circ \Phi_2)}{\partial x_{21}} \times \frac{\partial (\Psi \circ \Phi_2)}{\partial x_{22}} \right\| \mathrm{d}x_{21}\, \mathrm{d}x_{22}
\end{aligned} \tag{6.5}$$

where the magnitude of the cross product of the partial derivatives is known as the surface element. The above integrals can be regarded as component-wise surface integrals of scalar fields, yielding a set of 2 equations. Since the value of a surface

integral is independent of the parameterization, the above equality holds because both sides contain an integral on $\mathcal{S}_1$, parameterized through Φ_1 on the left hand side and through $\Psi \circ \Phi_2$ on the right hand side.

6.3.1 Construction of a System of Equations

Obviously, 2 equations are not enough to determine the 8 parameters of a homography. In order to generate more equations, let us remark that the identity relation in (6.4) remains valid when a function $\omega : \mathbb{R}^3 \to \mathbb{R}$ is acting on both sides of the equation (Domokos et al. 2012). Indeed, for a properly chosen ω

$$\omega(\mathbf{x}_{\mathcal{S}_1}) = \omega(\Psi(\Phi_2(\mathbf{x}_2))). \tag{6.6}$$

We thus obtain the following integral equation from (6.5) and (6.6)

$$\begin{aligned} \iint_{\mathcal{D}} \omega_i(\Phi_1(\mathbf{x}_1)) \left\| \frac{\partial \Phi_1}{\partial x_{11}} \times \frac{\partial \Phi_1}{\partial x_{12}} \right\| \mathrm{d}x_{11}\,\mathrm{d}x_{12} = \\ \iint_{\mathcal{F}} \omega_i(\Psi(\Phi_2(\mathbf{x}_2))) \left\| \frac{\partial(\Psi \circ \Phi_2)}{\partial x_{21}} \times \frac{\partial(\Psi \circ \Phi_2)}{\partial x_{22}} \right\| \mathrm{d}x_{21}\,\mathrm{d}x_{22} \end{aligned} \tag{6.7}$$

The basic idea of the proposed approach is to generate sufficiently many independent equations by making use of a set of nonlinear (hence linearly independent) functions $\{\omega_i\}_{i=1}^{\ell}$. Each ω_i generates a new equation yielding a system of ℓ independent equations. Note however, that the generated equations contain no new information, they simply impose new linearly independent constraints. Although arbitrary ω_i functions could be used, power functions are computationally favorable (Domokos et al. 2012). In our experiments, we adopted the following functions:

$$\omega_i(\mathbf{x}_{\mathcal{S}}) = x_1^{l_i} x_2^{m_i} x_3^{n_i}, \text{ with } 0 \le l_i, m_i, n_i \le 2 \text{ and } l_i + m_i + n_i \le 3 \tag{6.8}$$

These functions provide an overdetermined system of 15 equations of the form of (6.7), which can be solved in the *least squares sense* via a standard *Levenberg-Marquardt* (LM) algorithm. The solution to the system directly provides the parameters of the homography $\mathbf{H}$.

The computational complexity is largely determined by the calculation of the integrals in (6.7). Since both cameras are calibrated, Φ_1 and Φ_2 are known, hence the integrals on the left hand side are constant which need to be computed only once. However, the unknown homography $\mathbf{H}$ is involved in the right hand side through Ψ, hence these integrals have to be computed at each iteration of the LM solver. Of course, the spherical points $\mathbf{x}_{\mathcal{S}_2} = \Phi_2(\mathbf{x}_2)$ can be precomputed too, but the computation of the surface elements is more complex. First, let us rewrite the derivatives of

the composite function $\Psi \circ \Phi_2$ in terms of the Jacobian $\mathbf{J}_\Psi$ of Ψ and the gradients of Φ_2:

$$\left\| \frac{\partial(\Psi \circ \Phi_2)}{\partial x_{21}} \times \frac{\partial(\Psi \circ \Phi_2)}{\partial x_{22}} \right\| = \left\| \mathbf{J}_\Psi \frac{\partial \Phi_2}{\partial x_{21}} \times \mathbf{J}_\Psi \frac{\partial \Phi_2}{\partial x_{22}} \right\|$$

Since the gradients of Φ_2 are independent of $\mathbf{H}$, they can also be precomputed. Hence only $\Psi(\Phi_2(\mathbf{x}_2))$ and $\mathbf{J}_\Psi(\Phi_2(\mathbf{x}_2))$ have to be calculated during the LM iterations yielding a computationally efficient algorithm.

6.3.2 Normalization and Initialization

Since the system is solved by minimizing the algebraic error, proper normalization is critical for numerical stability (Domokos et al. 2012). Unlike in Domokos et al. (2012), spherical coordinates are already in the range of $[-1, +1]$, therefore no further normalization is needed. However, the ω_i functions should also be normalized into $[-1, +1]$ in order to ensure a balanced contribution of each equations to the algebraic error. In our case, this can be achieved by dividing the integrals with the maximal magnitude of the surface integral over the half unit sphere. We can easily compute these integrals by parameterizing the surface via points on the unit circle in the $x - y$ plane as $f(x, y) = (x, y, \sqrt{1 - x^2 - y^2})^T, \forall \|(x, y)\| < 1$. Thus the normalizing constant N_i for the equation generated by the function ω_i is

$$N_i = \iint\limits_{\|(x,y)\|<1} |\omega_i(f(x, y))| \sqrt{\frac{1}{1 - x^2 - y^2}} \, \mathrm{d}x \, \mathrm{d}y \tag{6.9}$$

To guarantee an optimal solution, initialization is also important. In our case, a good initialization ensures that the surface patches $\mathcal{D}_\mathcal{S}$ and $\mathcal{F}_\mathcal{S}$ overlap as much as possible. This is achieved by computing the centroids of the surface patches $\mathcal{D}_\mathcal{S}$ and $\mathcal{F}_\mathcal{S}$ respectively, and initializing $\mathbf{H}$ as the rotation between them.

The pseudo code of the proposed method is presented below.

Algorithm 6.8 The proposed homography estimation algorithm.

Input: A pair of 2D omnidirectional images with the same planar region segmented
Output: Homography $\mathbf{H}$ between the spherical images of the region
1: Back-project the 2D images onto the unit spheres using Φ_1 and Φ_2.
2: Construct the system of equations of (6.7) using the polynomial ω_i functions in (6.8).
3: Normalize the equations using (6.9)
4: Initialize the homography matrix $\mathbf{H}$ with the rotation between the centroids of the shapes on the sphere.
5: Solve the normalized nonlinear system of equations using the Levenberg-Marquardt algorithm.

6.4 Omnidirectional Camera Models

We have developed a homography estimation algorithm in Sect. 6.3, which is independent of the camera's internal projection functions Φ_1 and Φ_2. However, the knowledge of these functions as well as their gradient are necessary for the actual computation of the equations in (6.7). Herein, we will briefly overview two models that we used for experimental evaluation of the proposed method: the first one is the classical catadioptric camera model of Geyer and Daniilidis (2000) and the second one is the model of Scaramuzza (2006a) who derived a general polynomial form of the internal projection valid for any type of omnidirectional camera.

6.4.1 The General Catadioptric Camera Model

Let us first see the relationship between a 3D point $\mathbf{X}$ and its projection $\mathbf{x}$ in the omnidirectional image $\mathcal{I}$ (see Fig. 6.2). Note that only the half sphere on the image plane side is actually used, as the other half is not visible from image points. The camera coordinate system is in $\mathcal{S}$, the origin (which is also the center of the

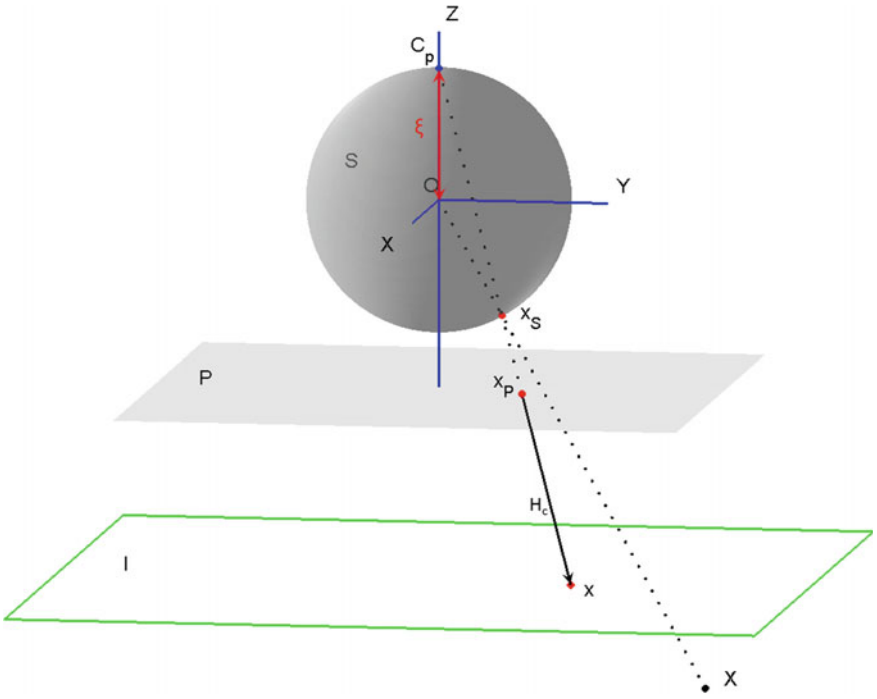

Fig. 6.2 Omnidirectional camera model using Geyer and Daniilidis' representation (Geyer and Daniilidis 2000)

sphere) is the *effective projection center* of the camera and the z axis is the optical axis of the camera which intersects the image plane in the *principal point*. To represent the nonlinear (but symmetric) distortion of central catadioptric cameras, (Geyer and Daniilidis 2000) projects a 3D point $\mathbf{X}$ from the camera coordinate system to a virtual projection plane $\mathcal{P}$ through the *virtual projection center* $\mathbf{C}_{\mathcal{P}} = (0, 0, -\xi)^T$ as

$$\mathbf{x}_{\mathcal{P}} = h(\mathbf{X}) = \begin{bmatrix} X_1 \\ X_2 \\ X_3 + \xi\sqrt{X_1^2 + X_2^2 + X_3^2} \end{bmatrix}$$

The virtual plane $\mathcal{P}$ is then transformed in the image plane $\mathcal{I}$ (see Fig. 6.2) through the homography $\mathbf{H}_C$ as

$$\begin{aligned} \mathbf{x} &= \mathbf{H}_C \mathbf{x}_{\mathcal{P}} \\ \mathbf{H}_C &= \mathbf{K}_C \mathbf{R} \mathbf{M}_C, \end{aligned}$$

where $\mathbf{K}_C$ includes the perspective camera parameters (taking the picture of the mirror), $\mathbf{R}$ is the rotation between camera and mirror, while $\mathbf{M}_C$ includes the mirror parameters—see (Geyer and Daniilidis 2000) for details. Herein, we will assume an ideal setting: no rotation (i.e. $\mathbf{R} = \mathbf{I}$) and a simple pinhole camera with focal length f and principal point (x_0, y_0) yielding

$$\mathbf{H}_C = \begin{bmatrix} f\eta & 0 & x_0 \\ 0 & f\eta & y_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \gamma & 0 & x_0 \\ 0 & \gamma & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where $\gamma = f\eta$ is the generalized focal length of the camera-mirror system and η is the mirror parameter. According to Geyer and Daniilidis (2000), this representation includes:

(1) conventional perspective cameras as $\xi = 0$ as well as
(2) catadioptric systems with parabolic mirror and orthographic camera for $\xi = 1$ and
(3) with hyperbolic mirror and perspective camera for $0 < \xi < 1$

In the following, without loss of generality, we will focus on case (2). The bijective mapping $\Phi : \mathcal{I} \to \mathcal{S}$ is the inverse of the camera's projection function, which is composed of (1) transforming the image point $\mathbf{x} \in \mathcal{I}$ back to the $\mathcal{P}$ virtual projection plane by $\mathbf{H}_C^{-1}$

$$\mathbf{x}_{\mathcal{P}} = \mathbf{H}_C^{-1} \mathbf{x},$$

and then (2) projecting back this point $(x_{\mathcal{P}1}, x_{\mathcal{P}2}, x_{\mathcal{P}3})^T$ from $\mathcal{P}$ to a 3D ray through the virtual projection center $\mathbf{C}_{\mathcal{P}}$ (assuming $\xi = 1$):

$$\begin{aligned}\mathbf{X} = h^{-1}(\mathbf{x}_{\mathcal{P}}) &= \begin{bmatrix} x_{\mathcal{P}1} \\ x_{\mathcal{P}2} \\ \frac{x_{\mathcal{P}3}^2 - x_{\mathcal{P}1}^2 - x_{\mathcal{P}2}^2}{2x_{\mathcal{P}3}} \end{bmatrix} \\ = h^{-1}(\mathbf{H}_C^{-1}\mathbf{x}) &= \begin{bmatrix} \frac{1}{\gamma}(x_1 - x_0) \\ \frac{1}{\gamma}(x_2 - y_0) \\ \frac{1}{2}\left(1 - \left(\frac{x_1 - x_0}{\gamma}\right)^2 - \left(\frac{x_2 - y_0}{\gamma}\right)^2\right) \end{bmatrix} \end{aligned} \tag{6.10}$$

We thus get the following expression for $\Phi : \mathcal{I} \to \mathcal{S}$:

$$\Phi(\mathbf{x}) = \mathbf{x}_{\mathcal{S}} = \frac{h^{-1}(\mathbf{H}_C^{-1}\mathbf{x})}{\|h^{-1}(\mathbf{H}_C^{-1}\mathbf{x})\|} \tag{6.11}$$

which provides the corresponding spherical point $\mathbf{x}_{\mathcal{S}} \in \mathcal{S}$. $\nabla\Phi$ is easily obtained from (6.10) and (6.11).

6.4.2 *Scaramuzza's Omnidirectional Camera Model*

Following (Scaramuzza et al. 2006a, b), we assume that the camera coordinate system is in $\mathcal{S}$, the origin is the effective projection center of the omnidirectional camera. To represent the nonlinear distortion of central omnidirectional optics, (Scaramuzza et al. 2006a, b) places a surface g between the image plane and the unit sphere $\mathcal{S}$, which is rotationally symmetric around z (see Fig. 6.3). The details of the derivation can be found in (Scaramuzza et al. 2006a, b). Herein, as suggested by (Scaramuzza et al. 2006b), we will use a fourth order polynomial $g(\|\mathbf{x}\|) = a_0 + a_2\|\mathbf{x}\|^2 + a_3\|\mathbf{x}\|^3 + a_4\|\mathbf{x}\|^4$ which has 4 parameters (a_0, a_2, a_3, a_4) representing the internal parameters of the camera (only 4 parameters as a_1 is always 0 according to Scaramuzza et al. (2006b)). The bijective mapping $\Phi : \mathcal{I} \to \mathcal{S}$ is composed of (1) lifting the image point $\mathbf{x} \in \mathcal{I}$ onto the g surface by an orthographic projection

$$\mathbf{x}_g = \begin{bmatrix} \mathbf{x} \\ a_0 + a_2\|\mathbf{x}\|^2 + a_3\|\mathbf{x}\|^3 + a_4\|\mathbf{x}\|^4 \end{bmatrix} \tag{6.12}$$

and then (2) centrally projecting the lifted point $\mathbf{x}_g$ onto the surface of the unit sphere $\mathcal{S}$:

$$\mathbf{x}_{\mathcal{S}} = \Phi(\mathbf{x}) = \frac{\mathbf{x}_g}{\|\mathbf{x}_g\|} \tag{6.13}$$

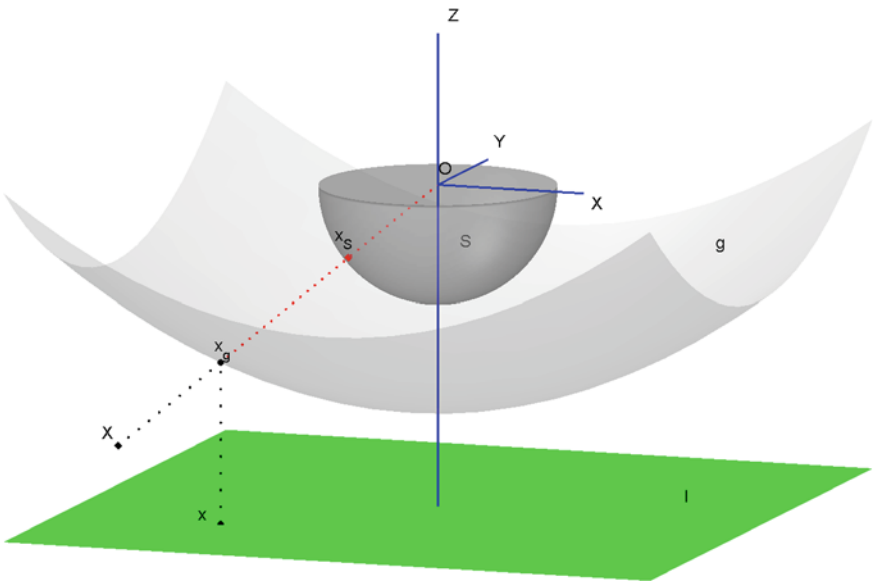

Fig. 6.3 Omnidirectional camera model using Scaramuzza's representation (Scaramuzza et al. 2006a, b)

Thus the omnidirectional camera projection is fully described by means of unit vectors $\mathbf{x}_S$ in the half space of $\mathbb{R}^3$ and these points correspond to the unit vectors of the projection rays. The gradient of Φ can be obtained from (6.12) and (6.13).

6.5 Experimental Results

A quantitative evaluation of the proposed method was performed by generating a total of 9 benchmark datasets, each containing 100 image pairs. Images of 24 different shapes were used as scene planes and a pair of virtual omnidirectional cameras with random pose were used to generate the omnidirectional images of 1MP. Assuming that these 800×800 scene plane images correspond to 5×5 m patches, we place the scene plane randomly at around 1.5 m in front of the first camera with a horizontal translation of ± 1 m and $\pm[5^\circ - 10^\circ]$ rotation around all three axis. The orientation of the second camera is randomly chosen having $\pm 5^\circ$ rotation around the x and y axis, and $\pm 10^\circ$ around the vertical z axis, while the location of the camera center is randomly chosen from the [45–55] cm, [100–200] cm, and [200–500] cm intervals, providing the first three datasets for 3 different baseline ranges. The alignment error (denoted by δ) was evaluated in terms of the percentage of non overlapping area of the omni images after applying the homography.

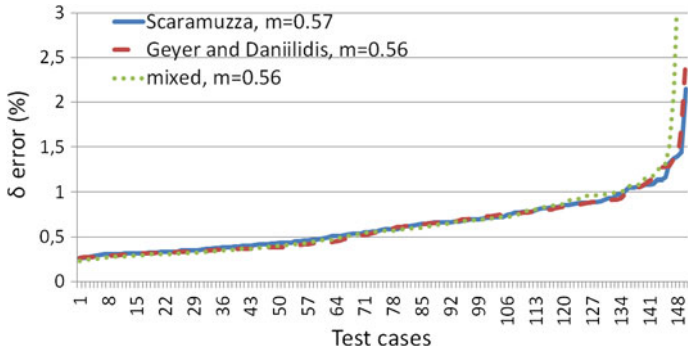

Fig. 6.4 Alignment (δ) error of the homography for various internal projection models (Scaramuzza et al. (2006a, b), Geyer and Daniilidis (2000), and mixed; m stands for median)

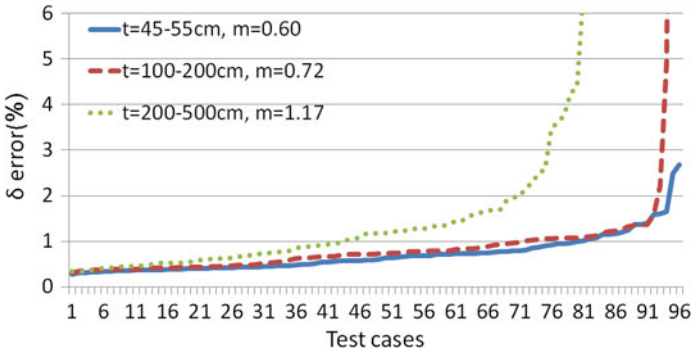

Fig. 6.5 Alignment error (δ) on the synthetic dataset with various baselines (m is the median, best viewed in color)

Based on our experimental results, we concluded that a registration error below 5 % corresponds to a correct alignment with a visually good matching of the shapes. For the synthetic datasets, error plots are shown in Figs. 6.4 and 6.5. Note that each plot represents the performed test cases sorted independently in a best-to-worst sense. In Fig. 6.4, we present a quantitative comparison of homography estimation with each of the camera models described in Sect. 6.4; as well as a test case with mixed cameras, where the first camera uses the Scaramuzza's polynomial representation and the second adopts the general catadioptric model. As expected, the quality of homography estimates is independent of the internal projection functions, both models perform well, error plots almost completely overlap. Therefore in all other test cases, we will only use Scaramuzza's model from Sect. 6.4.2.

The median value of δ was 0.60 %, 0.72 % and 1.17 % for the different baselines. In the first 2 cases, with baselines having values under 200 cm, we can say that only 1 % of the results were above 5 % error, while in the case of the biggest baselines

[200–500] cm still 84 % of the results are considered good, having δ error smaller than 5 %. The wrong results are typically due to extreme situations where the relative translation from the first camera to the second camera's position is in such a direction from where the image plane can be seen under a totally different angle resulting a highly different distortion of the shape on the omni image.

We have also tested the robustness of our method in some cases with unfavorable camera poses, see Fig. 6.6. One such situation is when the image of the actual planar region gets captured on the periphery of the omnidirectional image. It is well known, that these cameras have a much higher distortion in these regions. For this purpose we generated another synthetic dataset, making sure that all the regions fall on the periphery of the omnidirectional image. Another situation is when the relative camera pose has a much higher translation along the z axis, resulting a considerable size difference of the regions on the omnidirectional images. For this experiment a new synthetic dataset was generated with a bigger translation along the z axis (in the range of ± 1 m). The alignment errors of these two test cases are shown in Fig. 6.7. As we can see, the differences in the size of the regions that occur when having translation along the z axis are well tolerated by the algorithm, a homography can be estimated with almost the same precision. On the other hand, the higher distortion at the periphery of the images results in considerable loss of resolution, hence the homography estimation also looses some precision, but the median of the δ errors are still below 2 %. In summary, these results demonstrate that the method is robust against both unfavorable situations.

In practice, the shapes are segmented from real world images subject to various degree of segmentation errors. Therefore robustness against segmentation errors was also evaluated on simulated data. For this we used the dataset having the typical base

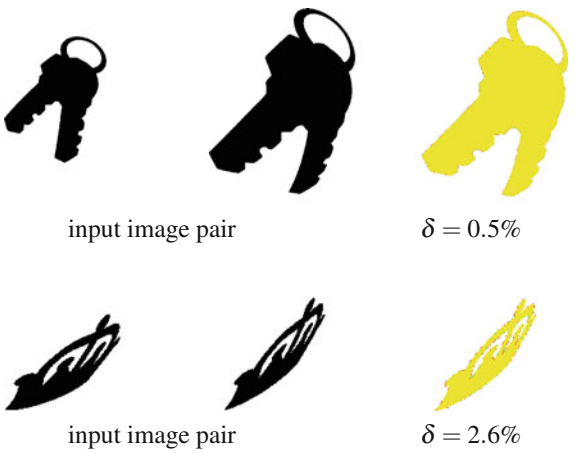

Fig. 6.6 Typical registration results for the test cases with unfavorable camera pose. First row shows a test case with big translation in the z, while the second row contains a test case with region falling on the periphery of the image

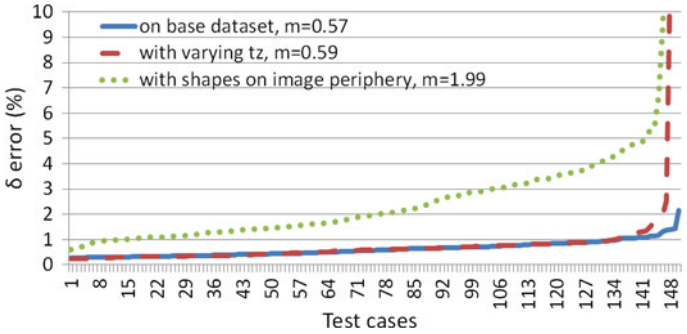

Fig. 6.7 Alignment error (δ) on the synthetic datasets with unfavorable camera poses (m is the median, best viewed in color)

distances of [1–2] m and we generated segmentation error by randomly adding and removing squares uniformly around the boundary of the shapes in one of the image pairs. A total of four datasets were produced from 5 % up to 20 % of boundary error. Samples from these datasets can be seen in Fig. 6.8, while Fig. 6.9 shows error plots for these datasets. Obviously, the median of δ error increases with the segmentation error, but the method shows robustness up to around 15 % error level. In particular, 80 % and 60 % of the first two cases are visually good, while only 44 % and 30 % of the cases are below the desired 5 % δ error for larger segmentation errors.

The algorithm was implemented in Matlab and all benchmarks were run on a standard quad-core desktop PC, resulting a typical runtime of 5–8 s without the code being optimized.

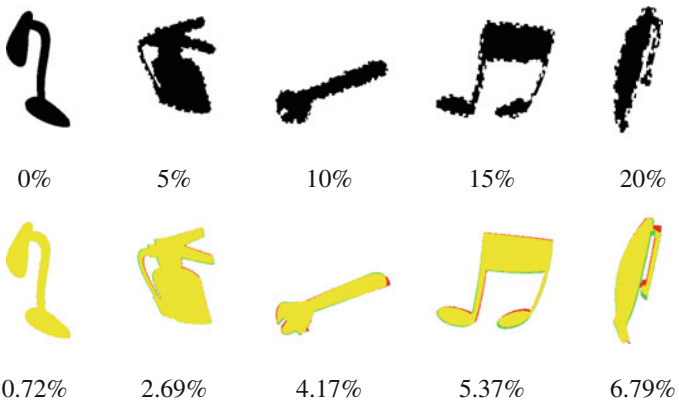

Fig. 6.8 Typical registration results for various level of segmentation error. First row shows the first image and the amount of segmentation error while the second row contains the overlay of the transformed second image over the first image with the δ error (best viewed in color)

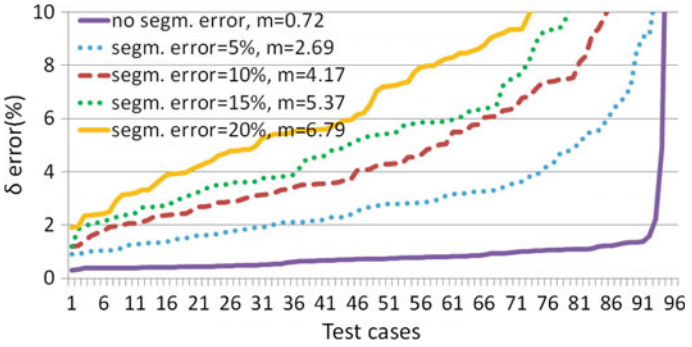

Fig. 6.9 Alignment error (δ) on the synthetic dataset with various levels of boundary error (m is the median, best viewed in color)

The real images, used for validation, were taken by a Canon 50D DSLR camera with a Canon EF 8–15 mm f/4L fisheye lens and the image size was 3MP. In our experiments, segmentation was obtained by simple region growing (initialized with only a few clicks) but more sophisticated and automatic methods could also be used. The extracted binary region masks where then registered by our method and the resulting homography has been used to project one image onto the other. Three such examples are illustrated in Fig. 6.10, where the first two images are the input omni image pairs, showing the segmented region in highlight, and the third image contains the transformed edges overlayed. We can observe that in spite of segmentation errors and slight occlusions (e.g. by the tree in the first image of Fig. 6.10), the edges of the reprojected region and the edges on the base image are well aligned. We should also mention that while slight occlusions are well tolerated, our method does not handle the occlusion of bigger parts of the region.

6.6 Relative Pose from Homography

If we consider again that the homography $\mathbf{H}$ is composed as in (6.2) from a rotation $\mathbf{R}$, the ratio $\mathbf{t}/d$ of the translation to the distance of plane and the normal $\mathbf{n}$ of the plane, we can express the pose parameters as described in Faugeras and Lustman (1988) using the singular value decomposition (SVD) of $\mathbf{H}$. Of course as the d distance of the plane is unknown, we can only express the translation $\mathbf{t}$ up to a scale factor. We fixed this scale factor by choosing the last element h_{33} of $\mathbf{H}$ to be 1.

The parameters that we obtain by the decomposition method can easily be verified in case of synthetic data, since we have the reference parameters saved during the dataset generation. The error in the relative translation can be characterized by either verifying the angle between the estimated and reference translation vectors, or by scaling up the estimated translation vector with the length of the reference translation

Fig. 6.10 Homography estimation results on real omni image pairs. Segmented regions are overlayed in lighter color, while the result is shown as the transformed green contours from the first image region over the second image

and computing the Euclidean distance between them. Here we have chosen to show the former one. The results can be seen in Fig. 6.11, where test cases are sorted by increasing δ error. We can observe that on a set of 150 test cases the estimated homography is really good, the δ error was below 2 % in all cases, and its median is less than 0.6 %. From a good input like this, the relative rotation and translation of the cameras can be factorized with high precision, only 0.19° median error in the rotation, and 0.51° in the direction of the translation vector.

The results show, that except a few test cases, the relative pose is determined with high stability. These few test cases (the spikes on Fig. 6.11) can be better explained by looking at Fig. 6.12 which shows only the factorized pose parameters for all test cases, sorted by the rotation error. The plot confirms a clear correlation between these

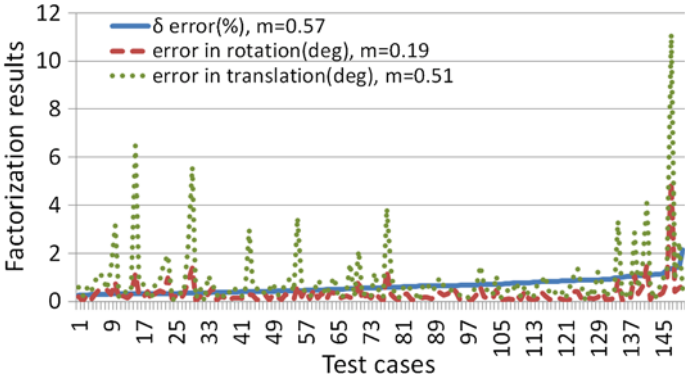

Fig. 6.11 Homography factorization results showing the δ error(%) of the homography, the rotation error and the translation error as the angle between the reference and factorized translation vectors (m is the median)

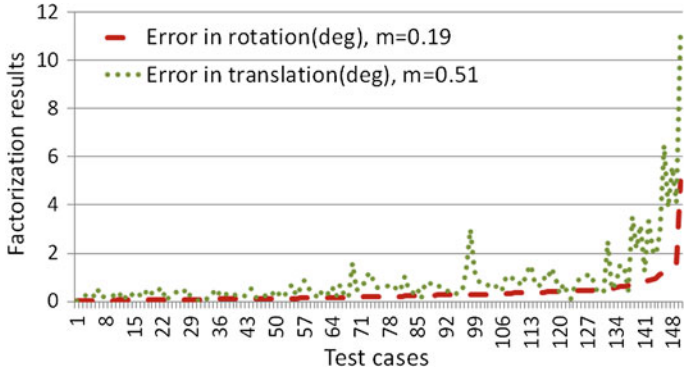

Fig. 6.12 Homography factorization results showing the rotation error and the translation error as the angle between the reference and factorized translation vectors, sorted by the rotation error (m is the median)

values, more visible on the second half of the plot, where the rotation and translation error increases together. This can be caused by the rare appearance of some specific camera configurations, where these errors in the parameters can compensate each other's effect, resulting in an overall good overlap (hence a low δ error) but spikes on Fig. 6.11.

Since the δ error of the homography in the previously mentioned dataset was considerably low (0.57 % of median error), we have also tested the factorization on the datasets with simulated segmentation error used in Sect. 6.5, where the homography errors span on a larger scale. The rotation error can be observed in Fig. 6.13. The effect of the worse homographies can obviously be seen on the factorized rotation, but still, at 10 % segmentation error, which resulted a δ error of 4.17 % for the dataset (see Fig. 6.9), the rotation error is well below 4° in median.

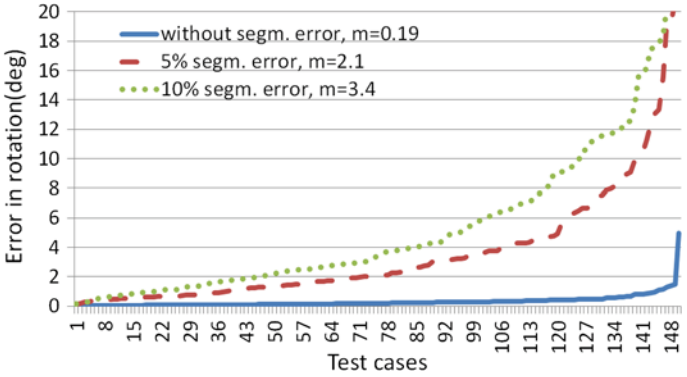

Fig. 6.13 Factorized rotation error with respect to different levels of segmentation error. Test cases sorted independently (m is the median)

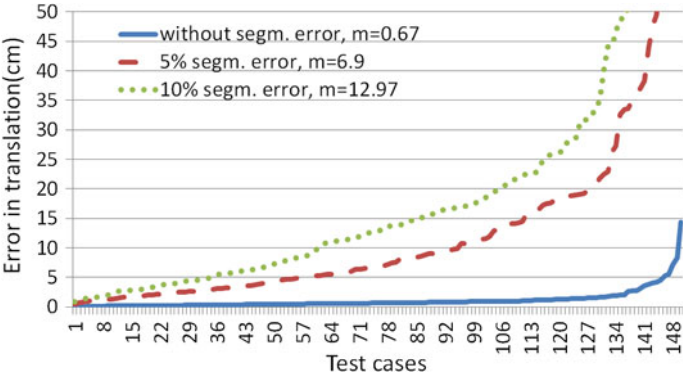

Fig. 6.14 Factorized translation error with respect to different levels of segmentation error. Test cases sorted independently (m is the median)

For the characterization of the translation errors in this case, we've expressed the Euclidean distance between the scaled up translation and the reference translation vector. The effect of the bigger δ error of the homographies in the different datasets can be observed in this case as well, visible in Fig. 6.14. The median of approximately 13 cm in the case of the 10 % segmentation error can be considered a reasonably good result, since our regions represent approximately 5×5 m surfaces in the scene.

Manhattan World Assumption

Manhattan world assumption is quite common when working with images of urban or indoor scenes (Coughlan and Yuille 1999; Furukawa et al. 2009). Although this is a strong restriction, yet it is satisfied at least partially in man-made structures. A somewhat relaxed assumption is the *weak Manhattan world* (Saurer et al. 2012) consisting of vertical planes with an arbitrary orientation but parallel to the gravity vector and orthogonal to the ground plane. Following (Saurer et al. 2012), we can also

take advantage of the knowledge of the vertical direction, which can be computed e.g. from an inertial measurement unit (IMU) attached to the camera. While (Saurer et al. 2012) deals with perspective cameras, herein we will show that homographies obtained from omnidirectional cameras can also be used and then we conduct a synthetic experiment to evaluate the performance of the method.

Let us consider a vertical plane π with its normal vector $\mathbf{n} = (n_x, n_y, 0)^T$ (z is the vertical axis, see Fig. 6.1). The distance d of the plane can be set to 1, because $\mathbf{H}$ is determined up to a free scale factor. Knowing the vertical direction, the rotation matrix $\mathbf{R}$ in (6.2) can be reduced to a rotation $\mathbf{R}_z$ around the z axis, yielding

$$\begin{aligned}\mathbf{H} &= \mathbf{R}_z + (t_x, t_y, t_z)(n_x, n_y, 0)^T \\ &= \begin{pmatrix} \cos(\alpha) + n_x t_x & -\sin(\alpha) + n_y t_x & 0 \\ \sin(\alpha) + n_x t_y & \cos(\alpha) + n_y t_y & 0 \\ n_x t_z & n_y t_z & 1 \end{pmatrix} \\ &= \begin{pmatrix} h_{11} & h_{12} & 0 \\ h_{21} & h_{22} & 0 \\ h_{31} & h_{32} & 1 \end{pmatrix}\end{aligned} \tag{6.14}$$

The estimation of such a *weak Manhattan* homography matrix is done in the same way as before, but the last column of $\mathbf{H}$ is set to $(0, 0, 1)^T$, yielding 6 free parameters only. In order to quantitatively characterize the performance of our method, 2 synthetic datasets with *weak Manhattan world* assumption were generated: first the 3D scene plane is positioned vertically and randomly rotated around the vertical axis by $[-10, +10]$ degrees, followed by a translation in the horizontal direction by $\pm$[400–800] pixels, equivalent to [2–4] m such that the surface of the plane is visible from the camera. For the second camera position we used a random rotation of $[-10, +10]$ degrees around the vertical axis followed by a horizontal translation of $\pm$[50–100] cm. The second dataset only differs in the vertical position of the 3D scene plane: in the first case, the plane is located approximately 150 cm higher than in the second case. Figure 6.15 shows the registration error for these datasets. As expected, having less free parameters increases estimation accuracy (alignment error is consistently under 2.5 %) and decreases computational time (typically 2–3 s).

Based on the above parameterization, $\mathbf{H}$ can be easily decomposed in the rotation α and the translation $\mathbf{t} = (t_x, t_y, t_z)^T$ parameters of the relative motion between the cameras. For example, using the fact that $n_x^2 + n_y^2 = 1, t_z = \pm\sqrt{h_{31}^2 + h_{32}^2}$ (see Saurer et al. (2012) for more details).

Following the decomposition method of Saurer et al. (2012), the horizontal rotation angle of the camera can be determined with a precision of around 0.6 degrees, which means a precision of a little above 5 % of the total rotation (see Fig. 6.16). As for the translation $\mathbf{t}$, it can be also recovered with an error of less then 5cm in the camera position. Note that the scale of $\mathbf{t}$ cannot be recovered from $\mathbf{H}$, but during the generation of our synthetic dataset, we also stored the length of the translation, hence we can use it to scale up the unit direction vector obtained from $\mathbf{H}$ and

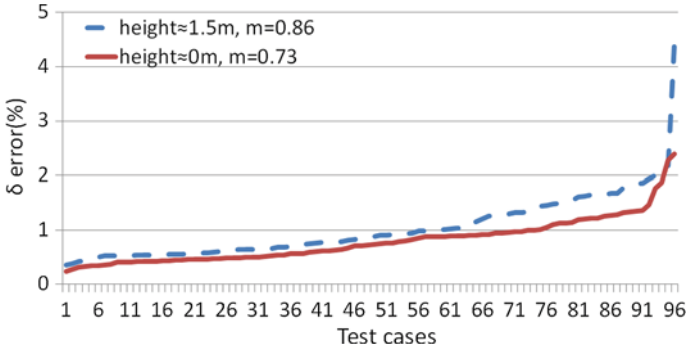

Fig. 6.15 Alignment error (δ) on the synthetic dataset with *weak Manhattan constraint* (only vertical surfaces and horizontal camera rotation allowed)

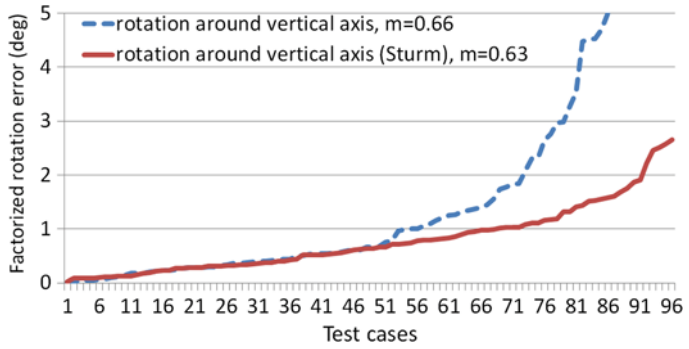

Fig. 6.16 Horizontal rotation error in relative pose (m is the median)

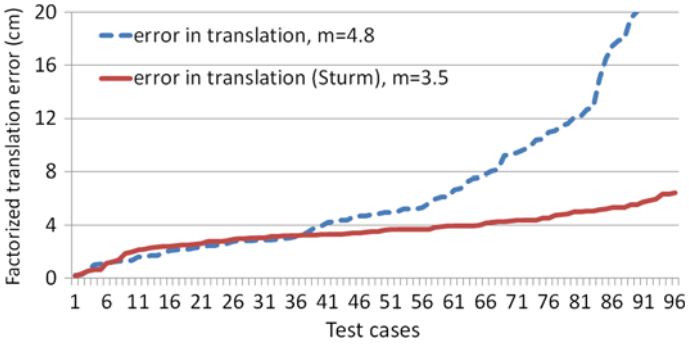

Fig. 6.17 Translation error in relative pose (m is the median)

compare directly the distance between the original and estimated camera centers. This is shown in the plots of Fig. 6.17.

Of course, classical homography decomposition methods could also be used. As an example, we show the pose estimation results obtained on the same dataset using the SVD-based factorization method from Sturm (2000). Figures 6.16 and 6.17 show the rotation and translation errors for both methods. Although the differences are not big, one can clearly see the increased stability of Sturm (2000).

6.7 Conclusions

In this chapter, a new homography estimation method has been proposed for central omnidirectional cameras. Unlike traditional approaches, we work with segmented regions corresponding to a 3D planar patch, hence our algorithm avoids the need for keypoint detection and descriptor extraction. In addition, being a purely shape-based approach, our method works with multimodal sensors as long as corresponding regions can be segmented in the different modalities. The parameters of the homography are directly obtained as the solution to a system of non-linear equations, whose size is independent of the input images. Furthermore, the method is also independent of the internal projection model of the camera as long as the projection function and its gradient are known. The algorithm is computationally efficient, allowing near-real time execution with a further optimized implementation. Quantitative evaluation on various synthetic datasets confirms the performance and robustness of the method under various conditions. We also demonstrate, that the accuracy of our homography estimates allows reliable estimation of extrinsic camera parameters.

Acknowledgments This research was partially supported by Domus MTA Hungary; and by the European Union and the State of Hungary, co-financed by the European Social Fund through projects FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013) and TAMOP-4.2.4.A/2-11-1-2012-0001 National Excellence Program. The authors would like to thank Levente Hajder for the Matlab implementation of the factorization method from Faugeras and Lustman (1988).

References

Baker S, Nayar SK (1999) A theory of single-viewpoint catadioptric image formation. Int J Comput Vis 35(2):175–196

Basri R, Jacobs DW (1996) Recognition using region correspondences. Int J Comput Vis 25:141–162

Caron G, Marchand E, Mouaddib EM (2011) Tracking planes in omnidirectional stereovision. In: IEEE international conference on robotics and automation. IEEE, pp 6306–6311

Coughlan J, Yuille AL (1999) Manhattan world: compass direction from a single image by bayesian inference. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999, vol 2, pp 941–947. doi:10.1109/ICCV.1999.790349

Domokos C, Nemeth J, Kato Z (2012) Nonlinear shape registration without correspondences. IEEE Trans Pattern Anal Mach Intell 34(5):943–958. doi:10.1109/TPAMI.2011.200

Faugeras O, Lustman F (1988) Motion and structure from motion in a piecewise planar environment. Technical Report RR-0856, INRIA, Sophia Antipolis, France. https://hal.inria.fr/inria-00075698

Furukawa Y, Curless B, Seitz S, Szeliski R (2009) Manhattan-world stereo. In: IEEE conference on computer vision and pattern recognition. IEEE Computer Society, Los Alamitos, pp 1422–1429. doi:10.1109/CVPRW.2009.5206867

Geyer C, Daniilidis K (2000) A unifying theory for central panoramic systems. In: European conference on computer vision (ECCV), pp 445–462

Gutierrez D, Rituerto A, Montiel J, Guerrero J (2011) Adapting a real-time monocular visual slam from conventional to omnidirectional cameras. In: 2011 IEEE international conference on computer vision workshops (ICCV Workshops), pp 343–350. doi:10.1109/ICCVW.2011.6130262

Hartley R, Zisserman A (2003) Multiple view geometry in computer vision, 2nd edn. Cambridge University Press, New York

Kannala J, Brandt SS (2006) A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. IEEE Trans Pattern Anal Mach Intell 28(8):1335–1340

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110

Makadia A, Geyer C, Daniilidis K (2007) Correspondence-free structure from motion. Int J Comput Vis 75(3):311–327. doi:10.1007/s11263-007-0035-2

Mei C, Rives P (2007) Single view point omnidirectional camera calibration from planar grids. IEEE international conference on robotics and automation (ICRA). Roma, Italy, pp 3945–3950

Mei C, Benhimane S, Malis E, Rives P (2008) Efficient homography-based tracking and 3-D reconstruction for single-viewpoint sensors. IEEE Trans Robot 24(6):1352–1364. doi:10.1109/TRO.2008.2007941

Mičušík B, Pajdla T (2004) Para-catadioptric camera auto-calibration from epipolar geometry. In: Hong KS, Zhang Z (eds) Proceedings of the asian conference on computer vision (ACCV). Asian Federation of Computer Vision Societies, Seoul, Korea South, vol 2, pp 748–753

Molnár J, Frohlich R, Dmitry C, Kato Z (2014a) 3D reconstruction of planar patches seen by omnidirectional cameras. Proceedings of international conference on digital image computing: techniques and applications. IEEE, Wollongong, Australia, pp 1–8

Molnár J, Huang R, Kato Z (2014b) 3D reconstruction of planar surface patches: A direct solution. In: Jawahar CV, Shan S (eds) Proceedings of ACCV workshop on big data in 3D computer vision, Lecture notes in computer science, vol 9008. Springer, Singapore, pp 286–300

Nayar SK (1997) Catadioptric omnidirectional camera. In: Proceedings of the 1997 conference on computer vision and pattern recognition (CVPR '97). IEEE Computer Society, Washington, USA, CVPR '97, pp 482–488. http://dl.acm.org/citation.cfm?id=794189.794460

Puig L, Guerrero JJ (2011) Scale space for central catadioptric systems: Towards a generic camera feature extractor. In: Proceedings of international conference on computer vision. IEEE, pp 1599–1606

Puig L, Guerrero JJ (2013) Omnidirectional vision systems: calibration, feature extraction and 3D information. Springer

Puig L, Bastanlar Y, Sturm P, Guerrero J, Barreto J (2011) Calibration of central catadioptric cameras using a DLT-Like approach. Int J Compu Vis 93(1):101–114. doi:10.1007/s11263-010-0411-1, https://hal.inria.fr/inria-00590268

Saurer O, Fraundorfer F, Pollefeys M (2012) Homography based visual odometry with known vertical direction and weak Manhattan world assumption. In: IEEE/IROS workshop on visual control of mobile robots (ViCoMoR)

Scaramuzza D, Martinelli A, Siegwart R (2006a) A flexible technique for accurate omnidirectional camera calibration and structure from motion. In: Proceedings of the fourth IEEE international conference on computer vision systems. IEEE Computer Society, Washington, USA, ICVS-06, pp 45–51

Scaramuzza D, Martinelli A, Siegwart R (2006b) A toolbox for easily calibrating omnidirectional cameras. In: Proceedings of the IEEE/RSJ international conference on intelligent robots. IEEE, Beijing, pp 5695–5701

Sturm P (2000) Algorithms for plane-based pose estimation. Proc Int Conf Comput Vis Pattern Recognit 1:706–711. doi:10.1109/CVPR.2000.855889

Sturm P, Ramalingam S, Tardif JP, Gasparini S, Barreto J (2011) Camera models and fundamental concepts used in geometric computer vision. Found Trends Comput Graph Vis 6(1–2):1–183. doi:10.1561/0600000023, https://hal.inria.fr/inria-00590269