Homography Estimation between Omnidirectional Cameras without Point Correspondences

Robert Frohlich[1*], Levente Tamas[2], and Zoltan Kato[1]

***Abstract*— This paper presents a novel approach for homography estimation between omnidirectional cameras. The solution is formulated in terms of a system of equations, where 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 under weak Manhattan assumption. The efficiency and robustness of the proposed method has been confirmed on both synthetic and real data.**

I. Introduction

Homography estimation is essential in many applications including pose estimation [1], tracking [2], [3], structure from motion [4] as well as recent robotics applications with focus on navigation [5], vision and perception. Efficient homography estimation methods exist for classical perspective cameras [6], but these methods are usually not reliable in case of omnidirectional sensors. The difficulty of homography estimation with omnidirectional 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 [7], [8], [9], [10], [11], [12]. The internal calibration of such cameras depends on these geometric models, which can be solved in a controlled environment, using special calibration patterns [11], [13], [14], [12]. 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 [2], [3]. 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. Classical keypoint detectors, such as SIFT [15], are also widely used [4], [2] 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 [16]. In [4], 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 feature-less image matching and recognition (most notably [17]), but without much success. In this paper, 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 [18]. The method has been quantitatively evaluated on a large synthetic dataset and proved to be robust and efficient. Inspired by [5], we also show that the estimated homography can be used to recover relative pose of an omnidirectional camera pair under the *weak Manhattan world* assumption.

II. Omnidirectional camera model

A unified model for central omnidirectional cameras was proposed by Geyer and Daniilidis [9], which represents central panoramic cameras as a projection onto the surface of a unit sphere. This formalism has been adopted and models for the internal projection function have been proposed by Micusik [10] and subsequently by Scaramuzza [19] who derived a general polynomial form of the internal projection

[1]Institute of Informatics, University of Szeged, Arpad ter 2, Hungary kato@inf.u-szeged.hu, frohlich@inf.u-szeged.hu

[2]Robotics Research Group, Technical University of Cluj-Napoca, Dorobantilor st. 73, 400609, Romania Levente.Tamas@aut.utcluj.ro

*On leave from Technical University of Cluj-Napoca.

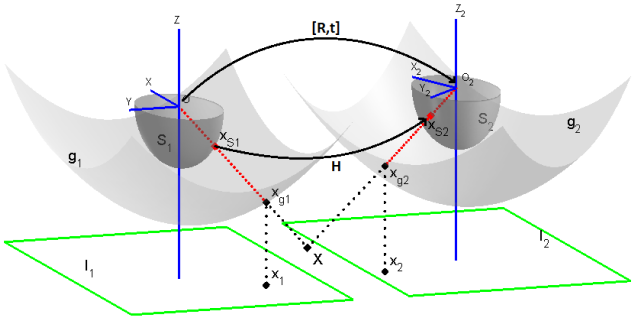


Fig. 1: Omnidirectional camera model

valid for any type of omnidirectional camera. In this work, we will use the latter representation.

Let us first see the relationship between a point $\mathbf{x}$ in the omnidirectional image $\mathcal{I}$ and its representation on the unit sphere $\mathcal{S}$ (see Fig. 1). Note that only the half sphere on the image plane side is actually used, as the other half is not visible from image points. Following [11], [19], we assume that the camera coordinate system is in $\mathcal{S}$, the origin (which is also the center of the sphere) is the 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 omnidirectional optics, [11], [19] places a surface g between the image plane and the unit sphere $\mathcal{S}$, which is rotationally symmetric around z. The details of the derivation of g can be found in [11], [19]. Herein, as suggested by [11], 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 [11]). 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

$$\mathbf{x}_g = \begin{bmatrix} \mathbf{x} \\ a_0 + a_2\|\mathbf{x}\|^2 + a_3\|\mathbf{x}\|^3 + a_4\|\mathbf{x}\|^4 \end{bmatrix} \tag{1}$$

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

$$\mathbf{x}_{\mathcal{S}} = \Phi(\mathbf{x}) = \frac{\mathbf{x}_g}{\|\mathbf{x}_g\|} \tag{2}$$

Thus the omnidirectional camera projection is fully described by means of unit vectors $\mathbf{x}_{\mathcal{S}}$ in the half space of $\mathbb{R}^3$ and these points correspond to the unit vectors of the projection rays.

A. Planar Homography

A 3D point $\mathbf{X} \in \mathbb{R}^3$ in the camera coordinate system is projected onto $\mathcal{S}$ by central projection. Therefore $\mathbf{X}$ and its image $\mathbf{x}$ in the omnidirectional camera are related as:

$$\Phi(\mathbf{x}) = \mathbf{x}_{\mathcal{S}} = \frac{\mathbf{X}}{\|\mathbf{X}\|} \tag{3}$$

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 (3), 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 $\mathbf{X}$ in the reference frame is

$$\mathbf{x}_{\mathcal{S}2} = \frac{\mathbf{R}\mathbf{X} + \mathbf{t}}{\|\mathbf{R}\mathbf{X} + \mathbf{t}\|} \tag{4}$$

Because of the single viewpoint, planar homographies stay valid for omnidirectional cameras too [2]. The standard planar homography $\mathbf{H}$ is composed up to a scale factor as

$$\mathbf{H} \propto \mathbf{R} + \frac{1}{d}\mathbf{t}\mathbf{n}^T \tag{5}$$

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. 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

$$\mathbf{X}_\pi = \lambda_1\mathbf{x}_{\mathcal{S}1} = \lambda_2\mathbf{H}\mathbf{x}_{\mathcal{S}2} \Rightarrow \mathbf{x}_{\mathcal{S}1} = \frac{\lambda_2}{\lambda_1}\mathbf{H}\mathbf{x}_{\mathcal{S}2} \tag{6}$$

Hence $\mathbf{x}_{\mathcal{S}1}$ and $\mathbf{H}\mathbf{x}_{\mathcal{S}2}$ are on the same ray yielding

$$\mathbf{x}_{\mathcal{S}1} = \frac{\mathbf{H}\mathbf{x}_{\mathcal{S}2}}{\|\mathbf{H}\mathbf{x}_{\mathcal{S}2}\|} = \Psi(\mathbf{x}_{\mathcal{S}2}) \tag{7}$$

III. Homography Estimation

Given a pair of omnidirectional cameras observing a planar surface, how to estimate the homography between their images? First, let us formulate the relation between a pair of corresponding omni image points $\mathbf{x}_1$, $\mathbf{x}_2$. According to (2), their lifted coordinates are obtained by applying the camera's inner projection functions Φ_1, Φ_2, and then the spherical points are related by (7):

$$\Phi_1(\mathbf{x}_1) = \mathbf{x}_{\mathcal{S}1} = \frac{\mathbf{H}\mathbf{x}_{\mathcal{S}2}}{\|\mathbf{H}\mathbf{x}_{\mathcal{S}2}\|} = \Psi(\Phi_2(\mathbf{x}_2)) \tag{8}$$

Any corresponding point pair $(\mathbf{x}_1, \mathbf{x}_2)$ satisfies the above equation. Thus a classical solution is to establish a set of such point matches by standard intensity-based point matching, and solve for $\mathbf{H}$. However, we are interested in a solution without point correspondences.

We will show that by identifying a single planar region in both images (denoted by $\mathcal{D}$ and $\mathcal{F}$), $\mathbf{H}$ can be estimated without any additional information. Since camera intrinsic parameters are known, we can work directly with spherical

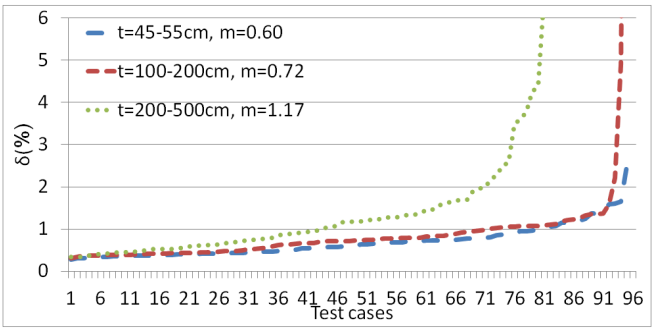


Fig. 2: Alignment error (δ) on the synthetic dataset with various baselines (m is the median, best viewed in color).

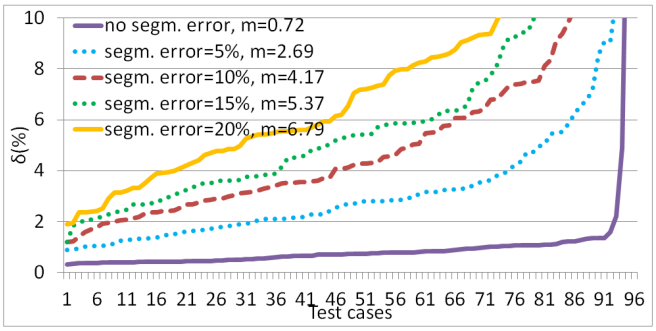


Fig. 3: Alignment error (δ) on the synthetic dataset with various levels of boundary error (m is the median, best viewed in color).

points. To get rid of individual point matches, we will integrate both sides of (8). This yields a surface integral on $\mathcal{S}_1$ over the surface patches $\mathcal{D}_\mathcal{S}$ and $\mathcal{F}_\mathcal{S}$ obtained by lifting the first omni image region $\mathcal{D}$ and by lifting and transforming the second omni image region $\mathcal{F}$. To get an explicit formula for these integrals, the surface patches $\mathcal{D}_\mathcal{S}$ and $\mathcal{F}_\mathcal{S}$ can be naturally parametrized via Φ_1 and $\Psi \circ \Phi_2$ over the planar regions $\mathcal{D}$ and $\mathcal{F}$: $\mathcal{D} \subset \mathbb{R}^2$, $\mathcal{F} \subset \mathbb{R}^2$; and $\forall \mathbf{x}_{\mathcal{S}_1} \in \mathcal{D}_\mathcal{S} : \mathbf{x}_{\mathcal{S}_1} = \Phi_1(\mathbf{x}_1), \mathbf{x}_1 \in \mathcal{D}$ as well as $\forall \mathbf{z}_{\mathcal{S}_1} \in \mathcal{F}_\mathcal{S} : \mathbf{z}_{\mathcal{S}_1} = \Psi(\Phi_2(\mathbf{x}_2)), \mathbf{x}_2 \in \mathcal{F}$, yielding the following integral equation:

$$\iint_\mathcal{D} \Phi_1(\mathbf{x}_1) \left\| \frac{\partial \Phi_1}{\partial x_{11}} \times \frac{\partial \Phi_1}{\partial x_{12}} \right\| \mathrm{d}x_{11}\,\mathrm{d}x_{12} = \iint_\mathcal{F} \Psi(\Phi_2(\mathbf{x}_2)) \left\| \frac{\partial(\Psi \circ \Phi_2)}{\partial x_{21}} \times \frac{\partial(\Psi \circ \Phi_2)}{\partial x_{22}} \right\| \mathrm{d}x_{21}\,\mathrm{d}x_{22} \tag{9}$$

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 $\mathcal{S}_1$, parametrized through Φ_1 on the left hand side and through $\Psi \circ \Phi_2$ on the right hand side.

A. Construction of a system of equations

Obviously, 2 equations are not enough to determine the 8 parameters of a homography. To construct a new set of equations, we adopt the general mechanism from [18] and apply a function $\omega : \mathbb{R}^3 \to \mathbb{R}$ to both sides of (8), yielding

$$\iint_\mathcal{D} \omega_i(\Phi_1(\mathbf{x}_1)) \left\| \frac{\partial \Phi_1}{\partial x_{11}} \times \frac{\partial \Phi_1}{\partial x_{12}} \right\| \mathrm{d}x_{11}\,\mathrm{d}x_{12} = \iint_\mathcal{F} \omega_i(\Psi(\Phi_2(\mathbf{x}_2))) \left\| \frac{\partial(\Psi \circ \Phi_2)}{\partial x_{21}} \times \frac{\partial(\Psi \circ \Phi_2)}{\partial x_{22}} \right\| \mathrm{d}x_{21}\,\mathrm{d}x_{22} \tag{10}$$

Adopting a set of nonlinear functions $\{\omega_i\}_{i=1}^{\ell}$, each ω_i generates a new equation yielding a system of ℓ independent equations. Although arbitrary ω_i functions could be used, power functions are computationally favorable [18]:

$$\omega_i(\mathbf{x}_\mathcal{S}) = x_1^{l_i} x_2^{m_i} x_3^{n_i}, \quad \text{with } 0 \le l_i, m_i, n_i \le 2 \text{ and } l_i + m_i + n_i \le 3 \tag{11}$$

Hence we are able to construct an overdetermined system of 15 equations, 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 $\mathbf{H}$.

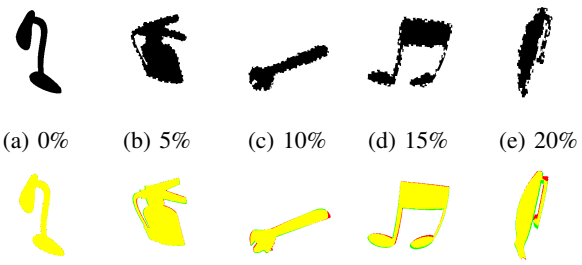
The computational complexity is largely determined by the calculation of the integrals in (10). Since both cameras are calibrated, Φ_1 and Φ_2 are known, hence the integrals on the left hand side are constant which need to be computed only once. However, the unknown homography $\mathbf{H}$ is involved 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:

$$\left\| \frac{\partial(\Psi \circ \Phi_2)}{\partial x_{21}} \times \frac{\partial(\Psi \circ \Phi_2)}{\partial x_{22}} \right\| = \left\| \mathbf{J}_\Psi \frac{\partial \Phi_2}{\partial x_{21}} \times \mathbf{J}_\Psi \frac{\partial \Phi_2}{\partial x_{22}} \right\| \tag{12}$$

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.

B. Normalization and Initialization

Since the system is solved by minimizing the algebraic error, proper normalization is critical for numerical stability [18]. Unlike in [18], 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



(a) 0% (b) 5% (c) 10% (d) 15% (e) 20%

(f) 0.72% (g) 2.69% (h) 4.17% (i) 5.37% (j) 6.79%

Fig. 4: 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 first image over the second image with the δ error (best viewed in color).

$f(x,y) = (x, y, \sqrt{1-x^2-y^2})^T, \forall \|(x,y)\| < 1$. Thus the normalizing constant N_i for the equation generated by the function ω_i is

$$N_i = \iint\limits_{\|(x,y)\|<1} |\omega_i(f(x,y))| \sqrt{\frac{1}{1-x^2-y^2}}\, \mathrm{d}x\, \mathrm{d}y \qquad (13)$$

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.

IV. Experimental Results

A quantitative evaluation of the proposed method was performed by generating a total of 9 benchmark datasets, each containing approximately 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 1MPx. Assuming that these 800×800 scene plane images correspond to 5×5m patches, we place the scene plane randomly at around 1.5m in front of the first camera with a horizontal translation of $\pm$1m and $\pm[5-10]$ degrees rotation around all three axis. The orientation of the second camera is randomly chosen having ± 5 degree around the x and y axis, and ± 10 degree around the vertical z axis, while the location of the camera center is randomly chosen from the [45cm-55cm], [100cm-200cm], and [200cm-500cm] 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. 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 Fig. 2. Note that each plot represents the performed test cases sorted independently in a best-to-worst sense. 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 200cm, we can say that only 1% of the results were above 5% error, while in the case of the biggest baselines (200cm to 500cm) 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, thus a hard task for the registration algorithm.

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 distances of [1m - 2m] 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. 4, while Fig. 3 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 the benchmarks were run on a standard quad-core desktop PC, resulting a typical runtime of 5 to 8 seconds with the code not being optimized in any way.

The real images, used for validation, were taken by a Canon 50D DSLR camera with a Canon EF 8-15mm f/4L fisheye lens and the image size was 3MPx. 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. Two such examples are illustrated in Fig. 5, 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. 5), the reprojected region's edges and the edges on the base image are well aligned.

V. Relative Pose from Homography

Manhattan world assumption is quite common when working with images of urban or indoor scenes [20], [21]. 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* [5] consisting of vertical planes with an arbitrary orientation but parallel to the gravity vector and orthogonal to the ground plane. Following [5], we can also take advantage of the knowledge

Fig. 5: 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.

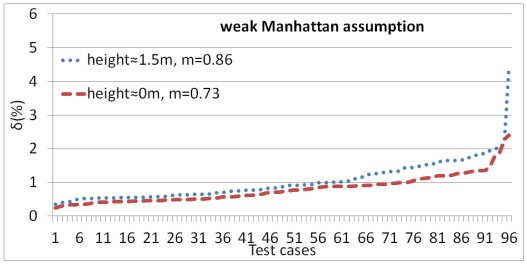


Fig. 6: Alignment error (δ) on the synthetic dataset with *weak Manhattan constraint* (only vertical surfaces and horizontal camera rotation allowed).

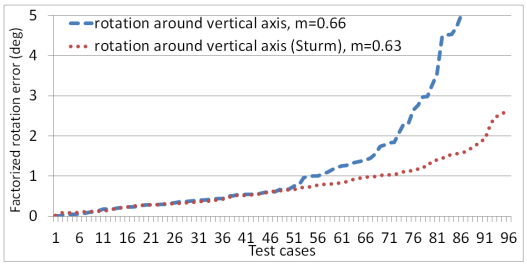


Fig. 7: Horizontal rotation error in relative pose (m is the median).

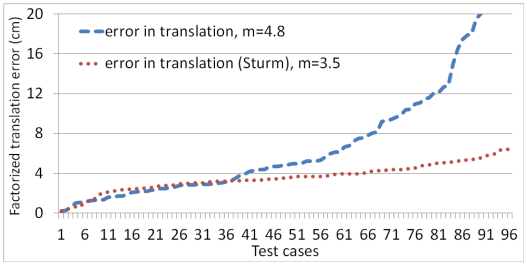


Fig. 8: Translation error in relative pose (m is the median).

of the vertical direction, which can be computed *e.g.* from an inertial measurement unit (IMU) attached to the camera. IMUs are widespread on modern smart phones. While [5] 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. 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 (5) can be reduced to a rotation $\mathbf{R}_z$ around the z axis, yielding

$$\begin{aligned}\mathbf{H} &= \mathbf{R}_z + (t_x, t_y, t_z)(n_x, n_y, 0)^T \\ &= \begin{pmatrix} \cos(\alpha) + n_x t_x & -\sin(\alpha) + n_y t_x & 0 \\ \sin(\alpha) + n_x t_y & \cos(\alpha) + n_y t_y & 0 \\ n_x t_z & n_y t_z & 1 \end{pmatrix} \\ &= \begin{pmatrix} h_{11} & h_{12} & 0 \\ h_{21} & h_{22} & 0 \\ h_{31} & h_{32} & 1 \end{pmatrix}\end{aligned} \quad (14)$$

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 [2m-4m] 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 $\pm$[50cm-100cm]. The second dataset only differs in the vertical position of the 3D scene plane: in the first case, the plane is located approximately 150cm higher than in the second case. Fig. 6

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 sec.).

Based on the above parametrization, $\mathbf{H}$ 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 [5] for more details).

Following the decomposition method of [5], 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. 7). 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 compare directly the distance between the original and estimated camera centers. This is shown in the plots of Fig. 8.

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 [1]. Fig. 7 and Fig. 8 show the rotation and translation errors for both methods. Although the differences are not big, one can clearly see the increased stability of [1].

VI. Conclusions

In this paper 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 is directly obtained as the solution to a system of non-linear equations, whose size is independent of the input images. 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 under *weak Manhattan world* assumption.

Acknowledgment

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.

References

[1] P. Sturm, "Algorithms for plane-based pose estimation," in *Proc. of International Conference on Computer Vision and Pattern Recognition*, vol. 1, June 2000, pp. 706–711.

[2] C. Mei, S. Benhimane, E. Malis, and P. Rives, "Efficient homography-based tracking and 3-D reconstruction for single-viewpoint sensors," *IEEE Trans. on Robotics*, vol. 24, no. 6, pp. 1352–1364, Dec. 2008.

[3] G. Caron, R. Marchand, and E. M. Mouaddib, "Tracking planes in omnidirectional stereovision." in *Proc. of International Conference on Robotics and Automation*. IEEE, 2011, pp. 6306–6311.

[4] A. Makadia, C. Geyer, and K. Daniilidis, "Correspondence-free structure from motion," *International Journal of Computer Vision*, vol. 75, no. 3, pp. 311–327, Dec. 2007.

[5] O. Saurer, F. Fraundorfer, and M. Pollefeys, "Homography based visual odometry with known vertical direction and weak Manhattan world assumption," in *IEEE/IROS Workshop on Visual Control of Mobile Robots (ViCoMoR)*, 2012.

[6] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.

[7] S. K. Nayar, "Catadioptric omnidirectional camera," in *Proc. of International Conference on Computer Vision and Pattern Recognition*. Washington, USA: IEEE Computer Society, 1997, pp. 482–.

[8] S. Baker and S. K. Nayar, "A theory of single-viewpoint catadioptric image formation," *International Journal of Computer Vision*, vol. 35, no. 2, pp. 175–196, 1999.

[9] C. Geyer and K. Daniilidis, "A unifying theory for central panoramic systems," in *Proc. of European Conference on Computer Vision*, 2000, pp. 445–462.

[10] B. Mičušík and T. Pajdla, "Para-catadioptric camera auto-calibration from epipolar geometry," in *Proc. of Asian Conference on Computer Vision*, K.-S. Hong and Z. Zhang, Eds., vol. 2. Seoul, Korea South: Asian Federation of Computer Vision Societies, January 2004, pp. 748–753.

[11] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras." in *Proc. of International Conference on Intelligent Robots*. Bejing: IEEE, October 9–15 2006, pp. 5695–5701.

[12] L. Puig and J. J. Guerrero, *Omnidirectional Vision Systems: Calibration, Feature Extraction and 3D Information*. Springer, 2013.

[13] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.

[14] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *Proc. of International Conference on Robotics and Automation*, Roma, Italy, April 2007.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] L. Puig and J. J. Guerrero, "Scale space for central catadioptric systems: Towards a generic camera feature extractor," in *Proc. of International Conference on Computer Vision*. IEEE, 2011, pp. 1599–1606.

[17] R. Basri and D. W. Jacobs, "Recognition using region correspondences," *International Journal of Computer Vision*, vol. 25, pp. 141–162, 1996.

[18] C. Domokos, J. Nemeth, and Z. Kato, "Nonlinear shape registration without correspondences," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 943–958, 2012.

[19] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A flexible technique for accurate omnidirectional camera calibration and structure from motion," in *Proc. of International Conference on Computer Vision Systems*, ser. ICVS-06. Washington, USA: IEEE Computer Society, 2006, pp. 45–51.

[20] J. Coughlan and A. L. Yuille, "Manhattan world: compass direction from a single image by Bayesian inference," in *Proc. of International Conference on Computer Vision*, vol. 2, 1999, pp. 941–947.

[21] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski, "Manhattan-world stereo," in *Proc. of International Conference on Computer Vision and Pattern Recognition*. Los Alamitos, CA, USA: IEEE Computer Society, 2009, pp. 1422–1429.