Doctoral School of Computer Science
Institute of Informatics
University of Szeged

Region-Based Pose and Homography Estimation for Central Cameras

Summary of the Ph.D. Thesis

by

Robert Frohlich

Supervisor:
Prof. Zoltan Kato

External Consultant:
Dr. Levente Tamas

Szeged
2019

Introduction

Computer vision is the scientific field that enables computers to gain high-level understanding from a single digital image or sequence of images. Practically it seeks to automate tasks that the human visual system can do. Main tasks include the acquiring, processing, analyzing and understanding of digital images, and extracting of information about the real world. The countless different applications available today can be enrolled in some well researched sub-domains like scene reconstruction, event detection, video tracking, object recognition, 3D pose estimation, learning, segmentation, motion estimation, and image restoration. The dissertation addresses the Author's research results in absolute pose estimation, homography estimation and planar scene reconstruction problems.

Region-based Pose Estimation

In this section we propose a generic, nonlinear, explicit correspondence-less pose estimation method. The absolute camera pose estimation is based on the 3D-2D registration of a common Lidar-camera planar patch. The proposed method makes use of minimal information (plain depth data from 3D and radiometric information from 2D) and is general enough to be used both for perspective and omnidirectional central cameras. The proposed framework is inspired by the 2D registration approach of [1], that was applied for a novel formulation of the absolute pose estimation of perspective cameras by [2], then further extended to omnidirectional cameras in [Tamas, Frohlich, Kato, 2014]. The general unifying framework proposed in [Frohlich, Tamas, Kato, 2019] uses a recursive formulation to compute the integrals in the system of equations, and compares it to the suboptimal scheme presented in [2]; extends the spherical representation of omni cameras from [Tamas, Frohlich, Kato, 2014] to include perspective cameras, and compares the two possible formulations for the perspective case; proposes a new numerical scheme for the surface integrals estimation over spherical triangles that is also compared to the earlier pixel-wise surface integration from [Tamas, Frohlich, Kato, 2014].

Let us formulate the absolute pose estimation problem for central spherical cameras. Considering the generalized spherical camera model proposed by [3, 4] we can clearly see

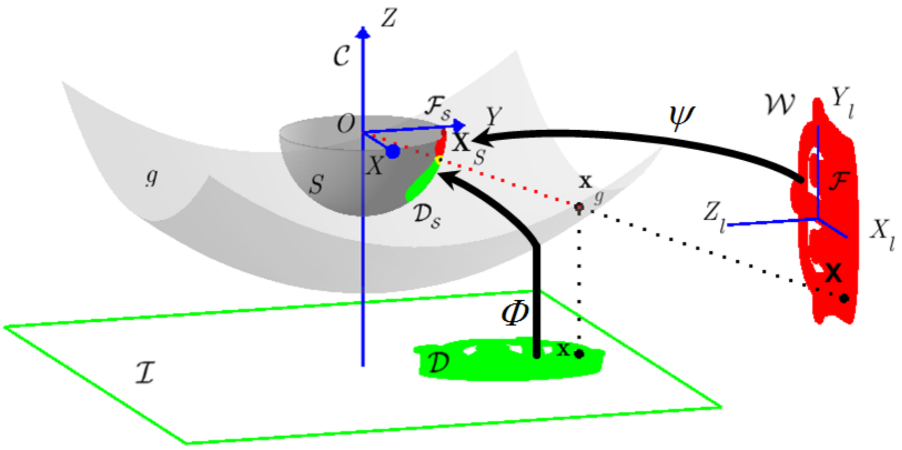

Figure 1. Spherical camera model and the projection of spherical patches $\mathcal{D}_S$ and $\mathcal{F}_S$.

that the projection of a 3D world point $\mathbf{X} = [X_1, X_2, X_3]^\top \in \mathbb{R}^3$ in the camera is basically a central projection onto $\mathcal{S}$ taking into account the extrinsic pose parameters ($\mathbf{R}$, $\mathbf{t}$). Thus for a world point $\mathbf{X}$ and its image $\mathbf{x} \in \mathcal{I}$, the following holds on the surface of $\mathcal{S}$ [Tamas, Frohlich, Kato, 2014]:

$$\Phi(\mathbf{x}) = \mathbf{X}_\mathcal{S} = \Psi(\mathbf{X}) = \frac{\mathbf{RX} + \mathbf{t}}{\|\mathbf{RX} + \mathbf{t}\|} \tag{1}$$

A classical solution of the absolute pose problem is to establish a set of 2D-3D point matches using *e.g.* a special calibration target [5, 6], or feature-based correspondences and then solve for $(\mathbf{R}, \mathbf{t})$ via the minimization of some error function based on (1). However, in many practical applications, it is not possible to use a calibration target and most 3D data (*e.g.* point clouds recorded by a Lidar device) will only record depth information, which challenges feature-based point matching algorithms. Therefore, we present a solution for such challenging situations.

Absolute Pose of Spherical Cameras

For spherical cameras, we have to work on the surface of the unit sphere as it provides a representation independent of the camera internal parameters. Furthermore, since correspondences are not available, (1) cannot be used directly. However, individual point matches can be integrated out yielding the following integral equation [Tamas, Frohlich, Kato, 2014]:

$$\iint_{\mathcal{D}_\mathcal{S}} \mathbf{X}_\mathcal{S} \, \mathrm{d}\mathcal{D}_\mathcal{S} = \iint_{\mathcal{F}_\mathcal{S}} \mathbf{Z}_\mathcal{S} \, \mathrm{d}\mathcal{F}_\mathcal{S}, \tag{2}$$

where $\mathcal{D}_\mathcal{S}$ denotes the surface patch on $\mathcal{S}$ corresponding to the region $\mathcal{D}$ visible in the camera image $\mathcal{I}$, while $\mathcal{F}_\mathcal{S}$ is the surface patch of the corresponding 3D planar region $\mathcal{F}$ projected onto $\mathcal{S}$ by Ψ in (1) as shown in Fig. 1.

To get an explicit formula for the above surface integrals, the spherical patches $\mathcal{D}_\mathcal{S}$ and $\mathcal{F}_\mathcal{S}$ can be naturally parametrized via Φ and Ψ over the planar regions $\mathcal{D}$ and $\mathcal{F}$ [Tamas, Frohlich, Kato, 2014]. Since a point on the surface $\mathcal{S}$ has only 2 independent components, this results a system of 2 equations only. Having 6 pose parameters we construct more equations by adopting the general mechanism from [1] and applying a function $\omega : \mathbb{R}^3 \to \mathbb{R}$ to both sides of the equation, yielding the following form of (2):

$$\iint_{\mathcal{D}} \omega(\Phi(\mathbf{x})) \left\| \frac{\partial \Phi}{\partial x_1} \times \frac{\partial \Phi}{\partial x_2} \right\| \mathrm{d}x_1 \, \mathrm{d}x_2 = \iint_{\mathcal{F}} \omega(\Psi(\mathbf{X})) \left\| \frac{\partial \Psi}{\partial X_1} \times \frac{\partial \Psi}{\partial X_2} \right\| \mathrm{d}X_1 \, \mathrm{d}X_2 \tag{3}$$

where the magnitude of the cross product of the partial derivatives is known as the surface element. 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. The pose parameters $(\mathbf{R}, \mathbf{t})$ are simply obtained by solving the nonlinear system of equations (3) in the *least squares sense* via a standard *Levenberg-Marquardt* algorithm.

Although arbitrary ω_i functions could be used, power functions are computationally favorable [1, 2] as these can be computed in a recursive manner:

$$\omega_i(\mathbf{X}_\mathcal{S}) = X_{S1}^{l_i} X_{S2}^{m_i} X_{S3}^{n_i}, \text{ with } 0 \leq l_i, m_i, n_i \leq 2 \text{ and } l_i + m_i + n_i \leq 3 \tag{4}$$

Note that the left hand side of (3) is constant, hence it has to be computed only once, but the right hand side involves the unknown pose parameters, thus has to be calculated in each iteration, which is computationally rather expensive. Therefore, in contrast to [Tamas, Frohlich, Kato, 2014] where the integrals on the 3D side were calculated over all points of the 3D region, let's consider a triangular mesh representation $\mathcal{F}^{\triangle}$ of the 3D planar region $\mathcal{F}$. Due to this representation, we only have to apply Ψ to the vertices $\{\mathbf{V}_i\}_{i=1}^{V}$ of the triangles in $\mathcal{F}^{\triangle}$, yielding a triangular representation [Frohlich, Tamas, Kato, 2019] of the spherical region $\mathcal{F}_{\mathcal{S}}^{\triangle}$ in terms of *spherical triangles*. The vertices $\{\mathbf{V}_{\mathcal{S},i}\}_{i=1}^{V}$ of $\mathcal{F}_{\mathcal{S}}^{\triangle}$ are obtained as

$$\forall i = 1, \ldots, V : \quad \mathbf{V}_{\mathcal{S},i} = \Psi(\mathbf{V}_i) \tag{5}$$

Due to this representation of $\mathcal{F}_{\mathcal{S}}$, we can rewrite the integral on the right hand side of (3) adopting ω_i from (4), yielding the following system of equations [Frohlich, Tamas, Kato, 2019]:

$$\iint\limits_{\mathcal{D}} \Phi_1^{l_i}(\mathbf{x})\Phi_2^{m_i}(\mathbf{x})\Phi_3^{n_i}(\mathbf{x}) \left\| \frac{\partial \Phi}{\partial x_1} \times \frac{\partial \Phi}{\partial x_2} \right\| \mathrm{d}x_1 \, \mathrm{d}x_2 \approx \sum_{\forall \triangle \in \mathcal{F}_{\mathcal{S}}^{\triangle}} \iint\limits_{\triangle} Z_{\mathcal{S}1}^{l_i} Z_{\mathcal{S}2}^{m_i} Z_{\mathcal{S}3}^{n_i} \, \mathrm{d}\mathbf{Z}_{\mathcal{S}}, \tag{6}$$

where $\Phi = [\Phi_1, \Phi_2, \Phi_3]^\top$ denote the coordinate functions of $\Phi : \mathcal{I} \to \mathcal{S}$. Thus only the triangle vertices need to be projected onto $\mathcal{S}$, and the integral over these spherical triangles is calculated using the method presented in [7]. The pose parameters are then obtained by solving the system of equations (6) in the least squares sense.

For an optimal estimate, it is important to ensure numerical normalization and a proper initialization. In contrast to [1], in the above equation all point coordinates are on the unit sphere, hence data normalization is implicit. To guarantee a good initialization of the pose parameters, multiple steps are performed. First, the 3D data is roughly aligned with our camera, ensuring that the camera is looking at the correct face of the surface in a correct orientation [Frohlich, Tamas, Kato, 2019]. Then we also apply a translation that brings the centroid of $\mathcal{F}^{\triangle}$ into $[0, 0, -1]^\top$, which puts the region into the $Z = -1$ plane. This is necessary to ensure that the plane doesn't intersect $\mathcal{S}$ while we initialize the pose parameters. If an approximate value for the vertical direction is available, which could be provided by different sensors, or the dataset itself, we roughly align the vertical direction to the camera's X axis, ensuring a correct vertical orientation of the projection. This might only be necesary when using symmetric regions. Initialization of the pose parameters ensures that the surface patches $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{F}_{\mathcal{S}}^{\triangle}$ overlap as much as possible [Frohlich, Tamas, Kato, 2019], by computing the centroids of $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{F}_{\mathcal{S}}^{\triangle}$, and initializing $\mathbf{R}$ as the rotation between them, and $\mathbf{t}$ as the translation of the planar region $\mathcal{F}^{\triangle}$ such that the area of $\mathcal{F}_{\mathcal{S}}^{\triangle}$ becomes approximately equal to that of $\mathcal{D}_{\mathcal{S}}$.

For two or more non-coplanar regions, the algorithm starts similarly, by first using only one region pair for an initial pose estimation, then starting from the obtained pose as an initial value, the system of equations is solved for all the available regions, which provides an overall optimal pose.

Absolute Pose of Perspective Cameras

As a central camera, the perspective camera can also be represented by the spherical camera model of [3, 4]. Since we assume a calibrated camera, we can define the projection of 3D world point $\mathbf{X}$ into normalized image coordinates $\mathbf{x} = [x_1, x_2]^\top \in \mathbb{R}^2$:

$$\mathbf{x} \leftarrow \mathbf{K}^{-1}\tilde{\mathbf{x}} \cong \mathbf{K}^{-1}\mathbf{P}\mathbf{X} = [\mathbf{R}|\mathbf{t}]\mathbf{X}, \tag{7}$$

Denoting the normalized image by $\mathcal{I}$, the surface g of the spherical model will be $g \equiv \mathcal{I}$, hence the bijective mapping $\Phi : \mathcal{I} \to \mathcal{S}$ for a perspective camera becomes simply the unit vector of $\mathbf{x}$:

$$\mathbf{X}_\mathcal{S} = \Phi(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|} \tag{8}$$

Based on the above spherical representation of a perspective camera, the whole method presented in the previous section applies without any change. However, it is computationally more favorable to work on the normalized image plane $\mathcal{I}$, because this way we can work with plain double integrals on $\mathcal{I}$ instead of surface integrals on $\mathcal{S}$. Hence applying a nonlinear function $\omega : \mathbb{R}^2 \to \mathbb{R}$ to both sides of (7) and integrating out individual point matches, we get [2]

$$\int_\mathcal{D} \omega(\mathbf{x})\, \mathrm{d}\mathbf{x} = \int_{[\mathbf{R}|\mathbf{t}]\mathcal{F}} \omega(\mathbf{z})\, \mathrm{d}\mathbf{z}. \tag{9}$$

where $\mathcal{D}$ corresponds to the region visible in the normalized camera image $\mathcal{I}$ and $[\mathbf{R}|\mathbf{t}]\mathcal{F}$ is the image of the corresponding *3D planar region* projected by the normalized camera matrix $[\mathbf{R}|\mathbf{t}]$. Choosing power functions for $\omega_i(\mathbf{x}) = x_1^{n_i} x_2^{m_i}$, and using a triangular mesh representation $\mathcal{F}^\triangle$ of the 3D region $\mathcal{F}$, we can adopt an efficient computational scheme, since this particular choice of ω_i yields equations, that contain the 2D geometric moments of the projected 3D region $[\mathbf{R}|\mathbf{t}]\mathcal{F}$. Therefore, we can rewrite the integral over $[\mathbf{R}|\mathbf{t}]\mathcal{F}^\triangle$ as [2]

$$\int_\mathcal{D} x_1^{n_i} x_2^{m_i}\, \mathrm{d}\mathbf{x} = \int_{[\mathbf{R}|\mathbf{t}]\mathcal{F}} z_1^{n_i} z_2^{m_i}\, \mathrm{d}\mathbf{z} \approx \sum_{\forall\triangle\in[\mathbf{R}|\mathbf{t}]\mathcal{F}^\triangle} \int_\triangle z_1^{n_i} z_2^{m_i}\, \mathrm{d}\mathbf{z}. \tag{10}$$

The latter approximation is due to the approximation of $\mathcal{F}$ by the discrete mesh $\mathcal{F}^\triangle$. The integrals over the triangles are various geometric moments which can be computed using efficient recursive formulas discussed hereafter.

Initialization of the pose parameters is done the same way as with the spherical cameras, except that for the pose parameters conditions are checked on the normalized image plane instead of the unit sphere. First a translation along the Z axis is determined such that the image region $\mathcal{D}$ and the projected 3D region are of the same size, then $\mathbf{R}$ is the rotation that brings the centroid of the projected 3D region close to the centroid of the corresponding image region $\mathcal{D}$ [Frohlich, Tamas, Kato, 2019].

2D Geometric Moments Calculation

For our method we need to calculate integrals over the regions $\mathcal{D} \subset \mathcal{I}$ and $[\mathbf{R}|\mathbf{t}]\mathcal{F}^\triangle \subset \mathcal{I}$, but we can easily adopt the efficient recursive formulas proposed [8] for geometric moments calculation over triangles in 3D and apply them to our 2D regions: Since our normalized image plane $\mathcal{I}$ is at $Z = 1$, the Z coordinate of the vertex points is a constant 1, hence the generic 3D formula for the (i, j, k) geometric moment of a surface S [8] becomes a plain 2D moment in our specific planar case [Frohlich, Tamas, Kato, 2019]:

$$M_{ijk} = \int_S x^i y^j z^k\, \mathrm{d}S = \int_S x^i y^j\, \mathrm{d}x\, \mathrm{d}y \tag{11}$$

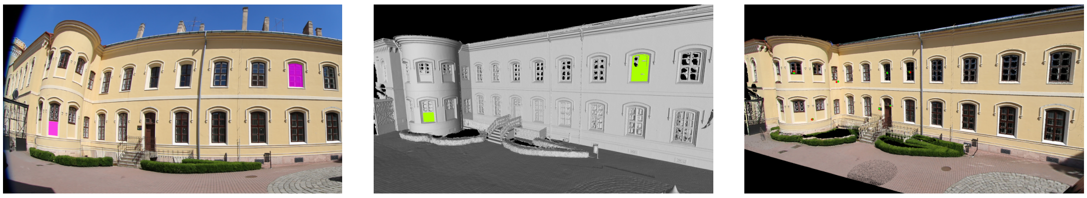

Figure 2. Pose estimation example with omni camera image (left) and dense Lidar data (middle), corresponding segmented regions are marked with purple and green respectively. Right: color information projected onto 3D data using the estimated extrinsic parameters (mean forward projection error on markers of 7 cm).

as the last term of M_{ijk} will always be 1 regardless of the value of k. i and j are integers such that $i + j = N$ is the order of the moment. Using the proposed equations in [Frohlich, Tamas, Kato, 2019], we can thus perform the exact computation of the contribution of every triangle to all the geometric moments of the image region in an efficient way.

The proposed algorithm has been implemented in MATLAB and was extensively tested on large scale synthetic datasets both for perspective and omnidirectional cameras. The method proved to be robust for up to 12% and 20% segmentation error in the omni and perspective case respectively, with 3 planar regions. It was also shown that increasing the number of planar regions from the minimal case of one single region, increases the performance drastically. Switching from pointwise representation to spherical triangles brings an order of magnitude improvement in the execution time of the algorithm. Multiple real tests were performed on data captured by sparse 3D laser scanners, dense Lidar scanners, catadioptric and fisheye cameras (see Fig. 2 for an example omni camera result), perspective DSLR, and drone cameras, some of the test cases having precise marker based reference parameters. Based on the publicly available KITTI [9] dataset, the proposed method proved comparable to the mutual information based method of [10], and the CPU implementation runtime was two orders of magnitude smaller than the GPU implementation of [10].

2D-3D Visual Data Fusion

In this section, we will present two visual data fusion applications. The first one builds on the perspective region-based registration method presented in [Frohlich, Tamas, Kato, 2019], extending the equations to non-planar but smooth surfaces, including an ICP like step for fine tuning the pose parameters if intensity information is available on the 3D data [Frohlich *et al.*, 2016]. The second application focuses on one of the key questions that arises when dealing with a large number of cameras, that is, the correct selection of views for colorizing the 3D model, and it also proposes a technical solution for visualizing the fusion results through textured models with a high number of texture images [Frohlich *et al.*, 2018].

Non-planar Regions

There are many applications that require a camera's absolute pose to be estimated, but in some cases the use of planar regions and a region-based method is not possible, simply because of the constraints of the 3D data. For example, in cultural heritage, there is an increasing demand for solutions to digitally document objects, locations or buildings, but in most of the cases, these objects, caves, ruins or old churches do not have suitable planar surfaces as a human built modern environment most probably would have.

Let us consider again the integral equation previously presented in (9). We can clearly see that the equation stays valid for curved, smooth surfaces as well [Frohlich *et al.*, 2016], as long as no self-occlusion of points takes place, thus $\mathcal{D}$ and $\mathcal{F}$ are satisfying:

$$\mathcal{D} = \mathbf{P}\mathcal{F}, \text{ with } \mathcal{D} = \cup_{i=1}^{N}\mathcal{D}_i \text{ and } \mathcal{F} = \cup_{j=1}^{M}\mathcal{F}_i \tag{12}$$

where $\{\mathcal{D}_i\}_{i=1}^{N}$ and $\{\mathcal{F}_j\}_{j=1}^{M}$ are a corresponding set of 2D-3D regions.

ICP Refinement

After obtaining a camera pose by minimizing the algebraic error of the system built in (10), we can further refine it by minimizing a relevant geometric error. In [Frohlich *et al.*, 2016] we have shown how a standard Iterative Closest Point (ICP) [11] algorithm can be used, if color information, even if it is of poor quality, is also available at each 3D point. In the proposed workflow, ICP is used to align the 3D edge lines' projection with the 2D edge map (denoted by $\mathbf{x_e}$) of the camera image. To ensure, that the same edges are detected in both domains, we simply project the 3D data onto an image with the initial camera pose, then detect edges on that image, resulting the 3D edge points $\mathbf{X_e}$. The algorithm then iteratively projects the 3D $\mathbf{X_e}$ edge points using the current $\mathbf{K}[\mathbf{R}^n|\mathbf{t}^n]$ camera matrix, that has only the camera pose parameters $(\mathbf{R}^n, \mathbf{t}^n)$ changing between iterations, giving the reprojected edge points $\mathbf{z_e}^n$ at iteration n:

$$\mathbf{z_e}^n = \mathbf{K}[\mathbf{R}^n|\mathbf{t}^n]\mathbf{X_e} \tag{13}$$

The ICP algorithm will align this $\mathbf{z_e}^n$ projection to $\mathbf{x_e}$, the edge map of the 2D image. Practically ICP will minimize the *backprojection error* this way.

Having the estimated relative pose and the calibration matrix of the camera, we can colorize the 3D points from the 2D image. If we had multiple 2D images, then for the 3D points visible in more camera images, we have to decide which camera has the best view of it. For this purpose we compute for every point $\mathbf{X}_i$ the angle of its normal $\mathbf{n}_i$ with the orientation vector $\mathbf{c}_j$ of each camera's optical axis as

$$\cos\theta = \frac{\mathbf{c_j} \cdot \mathbf{n_i}}{\|\mathbf{c_j}\|\|\mathbf{n_i}\|}, \tag{14}$$

and the camera image j with maximal $\cos\theta$ value is used to colorize the 3D point $\mathbf{X}_i$ [Frohlich *et al.*, 2016]. As a result, we get a good quality colored 3D model of the object. For smaller cultural heritage objects like ceramics or pottery fragments this kind of approach is sufficient, since a small number of views can cover the whole object, thus colorize the 3D model efficiently.

Quantitative evaluation on synthetic data proved, that the method performs well using regions on curved surfaces, even with one single region used, and it is robust to segmentation errors. Real data tests were performed on multiple interesting objects, both with high resolution images and precise 3D models, and with lower resolution images combined with 3D data captured by a hand held structured-light scanner. The latter case proved both the robustness of the proposed method and the low precision of these capturing devices.

Camera Selection

Let us consider now the cultural heritage applications that aim to document large caves, ruins, or old churches. Since scanning devices are getting more accessible and experts use them more often in their workflow, new solutions for data fusion are needed. Even though most of the commercial solutions provide a method to automatically register 2D camera images to the captured 3D model, these use the assumption of a rigidly mounted camera-scanner setup, not handling images taken from different views, or even in different time. Fusing the final 3D model is not solved well in most of the cases, thus the question of camera selection becomes more relevant in this scenario, if large number of images are used. We have proposed a full pipeline [Frohlich *et al.*, 2018] that employs a special algorithm for selecting the colorizing camera image for each 3D surface-element.

Practically the algorithm goes through a set of criteria, filtering out the *b*ad cameras in each step, and constructing a ranked list of the *g*ood cameras for each vertex. First we detect if a point $\mathbf{X}$ is visible from a camera or it is occluded. For this purpose, we have adopted the *Hidden Point Removal* operator [12]. It relies on the observation, that extracting the points that reside on the convex hull of a spherically flipped point cloud with respect to a given viewpoint, we get the visible points from that viewpoint.

Next, we verify if a point has a sharp image in the camera, so that only points that fall inside the *depth of field* of a camera C_i should be colorized from that camera image. The real world focus distance of a camera is not easily retrievable using only the image, but instead we can directly measure the upper and lower limits of the depth of field. Since for each image pixel we have the corresponding 3D point $\mathbf{X}$ we can directly compute the camera-to-point distance, we only have to find the image regions that are in focus. For this purpose, we adopt the focus measure introduced by [13], which reflects the statistical properties of the wavelet transform coefficients in different high frequency sub-bands. Using a sliding window technique, we select the windows w_s that have the focus measure above an experimentally determined threshold level, then simply calculate the average distance between the camera and the 3D points visible in window w_s as the average of the Euclidean distances from point to camera. Having a physical metric distance value $dist(w_s)$ assigned to each sharp window, we determine the lowest and highest distance limit by filtering out outliers. We apply these limits to filter out the cameras that don't see a given point sharply.

At this point, we have for each 3D point $\mathbf{X}$ a set of cameras assigned in which it's visible and in focus. As a next step we have to choose the one that sees the point from an optimal viewing angle and at highest resolution. Let us first calculate the angle between the surface normal $\mathbf{n_X}$ in $\mathbf{X}$ and the projection ray $\mathbf{o_{Xi}}$ pointing from $\mathbf{X}$ into the optical center of camera C_i. The angle of these two vectors can be simply calculated using:

$$\theta = \arccos\left(\frac{\mathbf{n_X} \cdot \mathbf{o_{Xi}}}{\|\mathbf{n_X}\| \cdot \|\mathbf{o_{Xi}}\|}\right) \tag{15}$$

with $\mathbf{o_{Xi}} = \mathbf{X} - \mathbf{c_i}$ being the projection vector of point $\mathbf{X}$ into the i^{th} camera. The angles $|\theta| \in (0 \ldots \pi/2)$ are the geometrically correct ones, as any other angle would mean that the camera is looking at the back side of the surface. Of course a mostly perpendicular view with small $|\theta|$ value is more favorable here.

Next, we also check the projection resolution of the region, since a higher focal length camera can produce higher level of detail even from a larger distance, or a lower focal length camera from a closer position as well might have better resolution. We characterize the resolution of the projection of point $\mathbf{X_m}$ in the i^{th} camera as $res_{mi} = f_i/D_{mi}$, where f_i is the focal length of the camera and D_{mi} is the distance of camera i from point $\mathbf{X_m}$.

Then the final decision is taken by choosing the camera with the highest value of

$$dc_{mi} = res_{mi}/\theta' \tag{16}$$

where θ' is the scaled version of angle θ into $\theta' \in [0 \ldots 1)$ with 0 corresponding to the perpendicular view and 1 corresponding to the $\pi/2$ angle. dc_{mi} stands for the decision value of camera i with respect to the 3D point $\mathbf{X_m}$.

Texture Mapping

In many applications it is desired to have reduced data size, while having the same visual resolution of the 3D model. This can be achieved by using a triangular mesh representation instead of the point cloud, and allows us to simplify the model by reducing the number of vertices, *e.g.* by decimation algorithms [14], and visualizing surfaces instead. This also brings the benefit that we can map texture to each triangle of the mesh.

Applying this to our proposed workflow [Frohlich *et al.*, 2018], we iterate over all the triangles F of the mesh instead of the points. This way we are able to select different cameras for neighboring faces that have common vertices, and we are not limited to one single camera assigned per vertex point. The camera ranking steps presented in the previous section still remain valid and necessary, we only have to adapt the final step of the algorithm, in this case iterating over faces F of the mesh. For each face we look at the three C^{v_k} camera ranking lists assigned to each vertex, that contains the previously defined dc decision values for all C_i cameras:

$$C^{v_k} = dc_{ki}, \text{ where } k \in (a, b, c) \text{ and } i \in (1..n) \tag{17}$$

and select the camera C_i that got included in all three C^{v_k} lists and has the highest values of dc. Assign this to face F_j:

$$C^{F_j} \in (C^{v_a} \cap C^{v_b} \cap C^{v_c}) \text{ where } dc = \max dc_{ki} \tag{18}$$

The data structure prepared this way can easily be written out in an ASCII Wavefront OBJ file based on its standard specifications [15].

The efficiency of the proposed method has been demonstrated on two large case studies. First, the documentation of the Reformed church of Somorja (Šamorín), then the documentation of the Reformed church in Kolozsnéma (Klížska Nemá), both of them located in Slovakia.

Homography Estimation

In this section, we will present our results on homography estimation related topics. Inspired by the 2D registration framework of [1], we have extended the approach to spherical cameras [Frohlich, Tamas, Kato, 2016]. Practically the homographies in this case act between the spherical projections in the two cameras, representing the image of the same planar region. In general, relative pose parameters [Frohlich, Tamas, Kato, 2016], as well as the normal [Molnár *et al.*, 2014] and distance of the inducing plane can be factorized from such a planar homography. But due to the inherent parametrization of a planar homography, direct approaches for solving the problem are also possible. We proposed a direct solution that simultaneously estimates the relative pose of the cameras, the planar homographies and the parameters of the inducing plane [Frohlich, Kato, 2018].

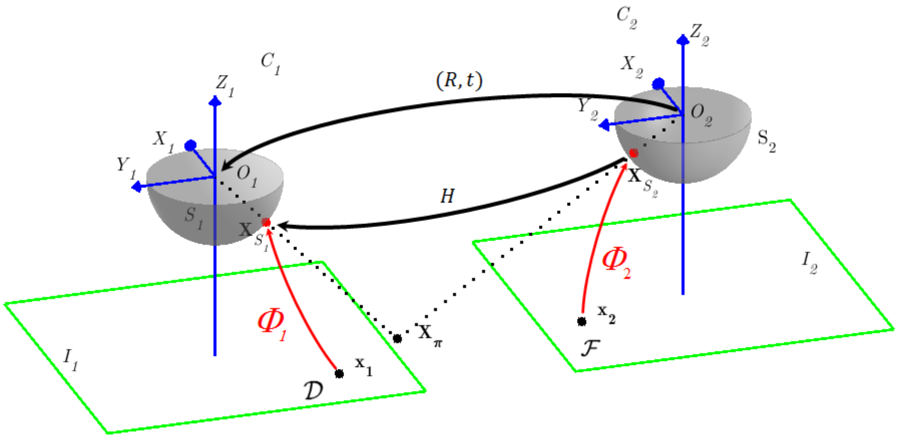

Figure 3. Homography acting between omnidirectional cameras represented as unit spheres.

Given a scene plane π, the mapping of plane points $\mathbf{X}_\pi \in \pi$ to the camera spheres $\mathcal{S}_i, i = 1, 2$ is governed by $\Phi(\mathbf{x}) = \mathbf{X}_\mathcal{S} = \frac{\mathbf{X}}{\|\mathbf{X}\|}$, hence it is bijective. 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, and the relative pose of the second camera is composed of a rotation $\mathbf{R}$ and translation $\mathbf{t} = (t_1, t_2, t_3)^T$, as shown in Fig. 3, thus projecting from sphere $\mathcal{S}_2$ to $\mathcal{S}_1$ is simply done by applying the same transformation, then normalizing the transformed point onto the unit sphere:

$$\mathbf{x}_{\mathcal{S}1} = \frac{\mathbf{R}\mathbf{X}_{\mathcal{S}2} + \mathbf{t}}{\|\mathbf{R}\mathbf{X}_{\mathcal{S}2} + \mathbf{t}\|}$$

Because of the single viewpoint, planar homographies stay valid for omni cameras too [16].

From our point of view, Φ provides an equivalent *spherical image* and the planar homography $\mathbf{H}$ simply acts between these spherical images [Frohlich, Tamas, Kato, 2016], as shown in Fig. 3. 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. Thus the spherical images $\mathbf{x}_{\mathcal{S}1}$, $\mathbf{x}_{\mathcal{S}2}$ of a point $\mathbf{X}_\pi$ on the plane and the corresponding omni image points $\mathbf{x}_1$ and $\mathbf{x}_2$ are related by

$$\Phi_1(\mathbf{x}_1) = \mathbf{X}_{\mathcal{S}1} = \frac{\mathbf{H}\mathbf{X}_{\mathcal{S}2}}{\|\mathbf{H}\mathbf{X}_{\mathcal{S}2}\|} = \Psi(\Phi_2(\mathbf{x}_2)) \tag{19}$$

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. Therefore we propose a solution without point matches, using regions. We integrate both sides of (19) 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 parametrized via Φ_1 and $\Psi \circ \Phi_2$ over the planar regions $\mathcal{D} \subset \mathbb{R}^2$ and $\mathcal{F} \subset \mathbb{R}^2$:

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

yielding the following integral equation:

$$\iint_{\mathcal{D}} \omega_i(\Phi_1(\mathbf{x}_1)) \left\| \frac{\partial \Phi_1}{\partial x_{11}} \times \frac{\partial \Phi_1}{\partial x_{12}} \right\| \mathrm{d}x_{11}\, \mathrm{d}x_{12} = \\ \iint_{\mathcal{F}} \omega_i(\Psi(\Phi_2(\mathbf{x}_2))) \left\| \frac{\partial(\Psi \circ \Phi_2)}{\partial x_{21}} \times \frac{\partial(\Psi \circ \Phi_2)}{\partial x_{22}} \right\| \mathrm{d}x_{21}\, \mathrm{d}x_{22} \quad (20)$$

In order to generate more equations, we have applied again the technique presented in [1]. Indeed, for a properly chosen ω

$$\omega(\mathbf{x}_{\mathcal{S}_1}) = \omega(\Psi(\Phi_2(\mathbf{x}_2))). \quad (21)$$

Thus we are able to generate sufficiently many independent equations by making use of a set of nonlinear (hence linearly independent) functions $\{\omega_i\}_{i=1}^{\ell}$. Note however, that the generated equations contain no new information, they simply impose new linearly independent constraints. The solution to the system directly provides the parameters of $\mathbf{H}$.

The computational complexity can largely be reduced by observing that the integrals on the left hand side of (20) are constant. 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 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. 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 each iteration yielding a computationally efficient algorithm [Frohlich, Tamas, Kato, 2016].

Since the system is solved by minimizing the algebraic error, proper normalization is critical for numerical stability [1]. Unlike in [1], 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. To guarantee an optimal solution, 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.

In this section we have presented a homography estimation algorithm, 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 (20). Robustness of the method was validated on synthetically generated data, while two of the most commonly used omnidirectional camera models were also compared.

Homography Factorization

Considering that a planar homography $\mathbf{H}$ is composed as

$$\mathbf{H} \propto \mathbf{R} - \mathbf{t}\mathbf{n}^T/d \tag{22}$$

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 [17] 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.

Another approach can be taken if we consider a man-made environment where the *weak Manhattan world* [18] assumption, consisting of vertical planes with an arbitrary orientation but parallel to the gravity vector and orthogonal to the ground plane, is satisfied. Following [18], we can 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 [18] deals with perspective cameras, we have shown that homographies obtained from omnidirectional cameras can also be used [Frohlich, Tamas, Kato, 2016].

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. 3). 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 (22) can be reduced to a rotation $\mathbf{R}_z$ around the z axis, yielding:

$$\mathbf{H} = \mathbf{R}_z - (t_x, t_y, t_z)(n_x, n_y, 0)^T \tag{23}$$

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 [Frohlich, Tamas, Kato, 2016]. 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 [Frohlich, Tamas, Kato, 2016]. For example, using the fact that $n_x^2 + n_y^2 = 1$, $t_z = \pm\sqrt{h_{31}^2 + h_{32}^2}$ (see [18] for more details).

Quantitative evaluation on synthetically generated *weak Manhattan* datasets proved, that both the homography estimation presented in the previous section, and the relative pose factorization from the estimated homographies can be performed robustly with the proposed methods, yielding comparable results to the standard SVD-based factorization method of [19].

Plane Reconstruction

In [Molnár *et al.*, 2014] a closed form solution was presented to reconstruct the normal vector of a 3D planar surface patch from the planar homography between a pair of corresponding image regions and known omnidirectional cameras, that was validated using the homography estimation method presented in the previous section. Once the normal vector $\mathbf{n}$ is determined, d can be easily computed based on (22) as shown *e.g.* in [20]. In the differential geometric solution of [Molnár *et al.*, 2014], the camera independent equations are constructed using the matrix-elements of the Jacobian of the linear transformations between image regions, expressed through the normal vector of the observed surface element and the gradients of the projection functions [21]:

$$[\mathbf{J}_{ij}] = \frac{1}{|\nabla x_i^1 \mathbf{n} \nabla x_i^2|} \begin{bmatrix} |\nabla x_j^1 \mathbf{n} \nabla x_i^2| & |\nabla x_i^1 \mathbf{n} \nabla x_j^1| \\ |\nabla x_j^2 \mathbf{n} \nabla x_i^2| & |\nabla x_i^1 \mathbf{n} \nabla x_j^2| \end{bmatrix} \tag{24}$$

The above quantities are all invariant first-order differentials: the gradients of the projections and the surface unit normal vector. Note that (24) is a general formula: neither a special form of projections, nor a specific surface is assumed here, hence it can be applied for any camera type and for any reasonably smooth surface. The formula can be used for computing the normal vector $\mathbf{n}$, when both the projection functions and the Jacobian $\mathbf{J}_{ij}$ are known. Let us write the matrix components of the derivatives of an estimated planar homography [Molnár *et al.*, 2014]:

$$[\mathbf{J}_{ij}]_{est} = \begin{bmatrix} a_1^1 & a_2^1 \\ a_1^2 & a_2^2 \end{bmatrix} \tag{25}$$

To eliminate the common denominator we can use ratios, which can be constructed using either row, column, or cross ratios [Molnár *et al.*, 2014]. Without loss of generality, the equation for the 3D surface normal can be deduced using cross ratios $\frac{a_1^1}{a_2^2}$ and $\frac{a_2^1}{a_1^2}$. After rearranging equation $[\mathbf{J}_{ij}]_{est} = [\mathbf{J}_{ij}]$ we obtain:

$$\begin{aligned} \mathbf{n} \cdot \left[a_2^2 \left(\nabla x_i^2 \times \nabla x_j^1\right) - a_1^1 \left(\nabla x_j^2 \times \nabla x_i^1\right)\right] =& 0 \\ \mathbf{n} \cdot \left[a_1^2 \left(\nabla x_j^1 \times \nabla x_i^1\right) - a_2^1 \left(\nabla x_i^2 \times \nabla x_j^2\right)\right] =& 0 \end{aligned} \tag{26}$$

Here we have two (known) vectors, both perpendicular to the normal:

$$\begin{aligned} \mathbf{p} =& \mathbf{n} \cdot \left[a_2^2 \left(\nabla x_i^2 \times \nabla x_j^1\right) - a_1^1 \left(\nabla x_j^2 \times \nabla x_i^1\right)\right] \\ \mathbf{q} =& \mathbf{n} \cdot \left[a_1^2 \left(\nabla x_j^1 \times \nabla x_i^1\right) - a_2^1 \left(\nabla x_i^2 \times \nabla x_j^2\right)\right] \end{aligned} \tag{27}$$

Thus the surface normal can readily be computed as

$$\mathbf{n} = \frac{\mathbf{p} \times \mathbf{q}}{|\mathbf{p} \times \mathbf{q}|}. \tag{28}$$

The normal vector can thus be expressed through the gradients of the projection functions. In [Molnár *et al.*, 2014] it was shown in detail how to compute the coordinate gradients $\nabla x_k^l, k = i, j; l = 1, 2$ w.r.t. spatial coordinates and $\mathbf{J}_{ij}$ in (24) for an omni camera pair.

In summary, given a pair of corresponding regions F and D in a pair of calibrated omnidirectional cameras with known projection functions Φ_i, Φ_j, the 3D scene plane π can be reconstructed through the following steps:

1. Estimate the homography $\mathbf{H}$ acting between the corresponding spherical regions $\mathcal{F}$ and $\mathcal{D}$ (using *e.g.* [Frohlich, Tamas, Kato, 2016]), which gives Ψ.

2. Estimate the relative pose $(\mathbf{R}, \mathbf{t})$ between the cameras. Given $\mathbf{H}$, this can be done by homography factorization methods, *e.g.* [19] or [Frohlich, Tamas, Kato, 2016].

3. Compute the normal $\mathbf{n}$ of π using the direct formula (28), and then d by a standard method based on (22), *e.g.* [20].

The algorithm was evaluated on synthetic data, and proved comparable performance to the well known classical plane from homography method described by Hartley and Zisserman [20].

Simultaneous Relative Pose and Plane Reconstruction

In contrast to the methods presented in the previous section, where a plane induced homography was first estimated between image regions, then the relative pose of the cameras was factorized, finally being able to calculate the parameters of the plane based on these estimated values, here we present a simultaneous solution for all the above problems for perspective cameras [Frohlich, Kato, 2018].

Starting from the absolute pose of perspective cameras, as described in the first section, we can work directly with the normalized images (7). Let us formulate the relation between a given scene plane π and its images $\mathcal{D}^0$ and $\mathcal{D}^1$ in two normalized cameras (see Fig. 4). Choosing the camera $\mathcal{C}_0$ as the reference frame, let us represent π by its unit normal $\mathbf{n} = (n_1, n_2, n_3)^\top$ and distance d to the origin. Furthermore, the relative pose of the second camera frame $\mathcal{C}_1$ is a 3D rigid body transformation $(\mathbf{R}^1, \mathbf{t}^1) : \mathcal{C}_0 \to \mathcal{C}_1$. Thus the image in the first and second camera of any homogeneous 3D point $\mathbf{X}$ of the reference frame is given by

$$\mathbf{x}_{\mathcal{C}_0} \cong [\mathbf{I}|\mathbf{0}]\mathbf{X} \quad \text{and} \quad \mathbf{x}_{\mathcal{C}_1} \cong [\mathbf{R}^1|\mathbf{t}^1]\mathbf{X}. \tag{29}$$

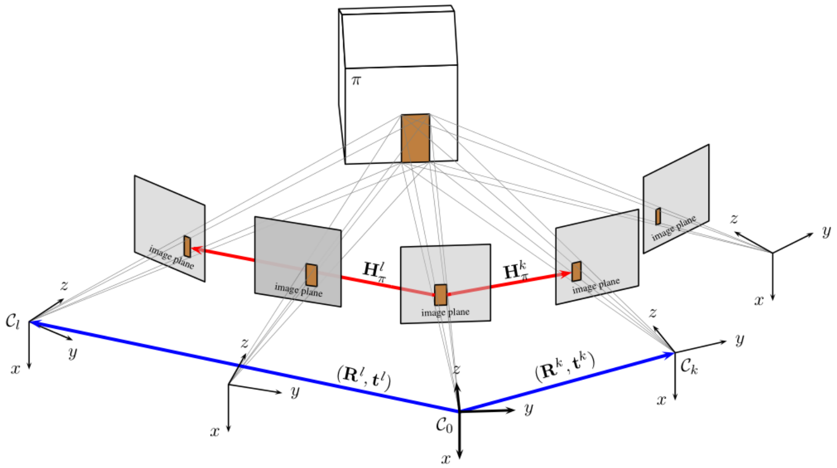

Figure 4. Projection of a 3D plane π in a multi-view camera system.

The mapping of 3D plane points $\mathbf{X}_\pi \in \pi$ into the cameras $\mathcal{C}_i, i = 0, 1$ is governed by the same equations, giving rise to a planar homography $\mathbf{H}^1_\pi : \mathcal{D}^0 \to \mathcal{D}^1$ induced by $\pi = (\mathbf{n}, d)$ between the image regions $\mathcal{D}^0$ and $\mathcal{D}^1$, composed up to a scale factor as (22). Thus for any point $\mathbf{X}_\pi \in \pi$, we have the following relation between the corresponding normalized image points $\mathbf{x}_{\mathcal{C}_0}$ and $\mathbf{x}_{\mathcal{C}_1}$:

$$\mathbf{x}_{\mathcal{C}_1} \cong \mathbf{H}^1_\pi \mathbf{x}_{\mathcal{C}_0} \cong (\mathbf{R}^1 - \frac{1}{d}\mathbf{t}^1\mathbf{n}^\top)\mathbf{x}_{\mathcal{C}_0}. \tag{30}$$

The classical solution is to find at least 4 such point matches and solve for $\mathbf{H}^1_\pi$, then factorize $\mathbf{R}^1$, $\mathbf{t}^1$, and $\mathbf{n}$ from $\mathbf{H}^1_\pi$ (d cannot be recovered due to the free scaling factor) [22]. However, our region-based approach [Frohlich, Kato, 2018] robustly recovers the alignment of non-linear shape deformations via the solution of a special system of equations without established point correspondences.

Following the idea of [1], we avoid working with point correspondences by integrating out both sides of (30). Applying an appropriate set of $\omega : \mathbb{R}^2 \to \mathbb{R}$ functions on both sides, the equality remains valid, yielding the following integral equation:

$$\int_{\mathcal{D}^1} \omega(\mathbf{x}_{\mathcal{C}_1})\,\mathrm{d}\mathbf{x}_{\mathcal{C}_1} = \int_{\mathcal{D}^0} \omega(\mathbf{H}^1_\pi \mathbf{x}_{\mathcal{C}_0}) |\mathbf{J}_{\mathbf{H}^1_\pi}(\mathbf{x}_{\mathcal{C}_0})|\,\mathrm{d}\mathbf{x}_{\mathcal{C}_0}. \tag{31}$$

where the integral transformation $\mathbf{x}_{\mathcal{C}_1} = \mathbf{H}^1_\pi \mathbf{x}_{\mathcal{C}_0}$, $\mathrm{d}\mathbf{x}_{\mathcal{C}_1} = |\mathbf{J}_{\mathbf{H}^1_\pi}(\mathbf{x}_{\mathcal{C}_0})|\,\mathrm{d}\mathbf{x}_{\mathcal{C}_0}$ has been applied. Since $\mathbf{H}^1_\pi$ is a 3×3 homogeneous matrix with only 8 DoF, we will set its last element to 1. Note that the above equality is true for inhomogeneous point coordinates $\mathbf{x}_{\mathcal{C}_i}$, which are obtained by projective division. The Jacobian determinant $|\mathbf{J}_{\mathbf{H}^1_\pi}| : \mathbb{R}^2 \to \mathbb{R}$ gives the measure of the transformation at each point [1].

The unknown relative pose $(\mathbf{R}^1, \mathbf{t}^1)$ and 3D plane parameters $(\mathbf{n}, d)$ are then simply obtained as the solution of the nonlinear system of equations (31). In practice, an overdetermined system is constructed, which is then solved in the *least squares sense* by minimizing the algebraic error via a standard *Levenberg-Marquardt* algorithm.

Multiple Regions and Multiple Views

The key advantage of the proposed solution when compared to classical homography estimation methods, is the possibility to handle in the same system multiple planar regions and/or cameras. Practically, since each plane π_i generates a homography $\mathbf{H}^1_{\pi_i}$ between the corresponding image regions $\mathcal{D}^0_i$ and $\mathcal{D}^1_i$, (30) and (31) remains valid for each of these homographies, but we have to note that the relative pose $(\mathbf{R}^1, \mathbf{t}^1)$ of the cameras is the same for all $\mathbf{H}^1_{\pi_i}$, they only differ in the 3D plane parameters $(\mathbf{n}_i, d_i)$. Hence for all $\{\pi_i\}_{i=1}^N$, we have

$$\mathbf{x}_{\mathcal{C}_1} \cong \mathbf{H}^1_{\pi_i} \mathbf{x}_{\mathcal{C}_0} \cong (\mathbf{R}^1 - \frac{1}{d_i}\mathbf{t}^1 \mathbf{n}_i^\top)\mathbf{x}_{\mathcal{C}_0}, \text{ with } \mathbf{x}_{\mathcal{C}_0} \in \mathcal{D}^0_i \text{ and } \mathbf{x}_{\mathcal{C}_1} \in \mathcal{D}^1_i \tag{32}$$

and (31) becomes a system of N equations [Frohlich, Kato, 2018] in terms of the common camera pose $(\mathbf{R}^1, \mathbf{t}^1)$ and the parameters $(\mathbf{n}_i, d_i)$ of the 3D planes $\{\pi_i\}_{i=1}^N$:

$$\int_{\mathcal{D}^1_i} \omega(\mathbf{x}_{\mathcal{C}_1})\,\mathrm{d}\mathbf{x}_{\mathcal{C}_1} = \int_{\mathcal{D}^0_i} \omega(\mathbf{H}^1_{\pi_i} \mathbf{x}_{\mathcal{C}_0}) |\mathbf{J}_{\mathbf{H}^1_{\pi_i}}(\mathbf{x}_{\mathcal{C}_0})|\,\mathrm{d}\mathbf{x}_{\mathcal{C}_0}, \quad 1 \le i \le N \tag{33}$$

For a given ω function, the above equations provide N constraints on the relative pose parameters, but only 1 constraint for each plane π_i, having a total of N equations. Note also, that we have one free scaling factor for the whole system in (33), because a relative d_i parameter for the planes need to be determined, only one of them can be set freely.

Similarly, when multiple cameras are observing the scene planes, then we can construct a system of equations which contains multiple constraints not only for the camera relative poses but also for each 3D plane.

A reference camera frame $\mathcal{C}_0$ is chosen, each camera's relative pose is determined w.r.t. $\mathcal{C}_0$ and all planes are reconstructed within $\mathcal{C}_0$. Assuming that all scene planes $\{\pi_i\}_{i=1}^N$ are visible in every camera $\{\mathcal{C}_k\}_{k=0}^{M-1}$, each plane π_i generates a homography $\mathbf{H}^k_{\pi_i}$ between the corresponding image regions in the reference camera $\mathcal{D}^0_i$ and the k^{th} camera $\mathcal{D}^k_i$:

$$\forall 1 \le k \le M-1 : \quad \mathbf{x}_{\mathcal{C}_k} \cong \mathbf{H}^k_{\pi_i} \mathbf{x}_{\mathcal{C}_0} \cong (\mathbf{R}^k - \frac{1}{d_i}\mathbf{t}^k \mathbf{n}_i^\top)\mathbf{x}_{\mathcal{C}_0}. \tag{34}$$

Hence each camera provides a new constraint on the scene plane parameters $(\mathbf{n}_i, d_i)$, yielding a total of $M - 1$ constraints for reconstructing π_i [Frohlich, Kato, 2018]. If a particular plane is not visible in all other cameras, then the number of these constraints is reduced.

A particular camera pair $(\mathcal{C}_0, \mathcal{C}_k)$ provides N equations in terms of the common camera pose $(\mathbf{R}^k, \mathbf{t}^k)$ and the parameters $(\mathbf{n}_i, d_i)$ of the 3D planes $\{\pi_i\}_{i=1}^N$, yielding a system of N equations similar to (31). Therefore we get

$$\int_{\mathcal{D}_i^k} \omega(\mathbf{x}_{\mathcal{C}_k})\,\mathrm{d}\mathbf{x}_{\mathcal{C}_k} = \int_{\mathcal{D}_i^0} \omega(\mathbf{H}_{\pi_i}^k \mathbf{x}_{\mathcal{C}_0}) |\mathbf{J}_{\mathbf{H}_{\pi_i}^k}(\mathbf{x}_{\mathcal{C}_0})|\,\mathrm{d}\mathbf{x}_{\mathcal{C}_0}, \qquad 1 \leq i \leq N \text{ and } 1 \leq k \leq M-1 \quad (35)$$

For a given ω function, the above equations provide N constraints on each relative pose $(\mathbf{R}^k, \mathbf{t}^k)$, and $M - 1$ constraints for each plane π_i, having a total of $N(M - 1)$ equations. The minimal number of equations needed to solve for $M \geq 2$ cameras and $N \geq 1$ planes is $E = 6(M - 1) + 3N - 1$.

When used with more cameras or more regions than the minimal setup assumes, the proposed method [Frohlich, Kato, 2018] applies a two step algorithmic approach. First, each neighboring camera pair is solved in a pairwise way, where the pose and plane parameters don't require any specific initialization. Then a final bundle adjustment step is performed, where all previous results are transformed into the chosen reference frame, relative poses are written up, and the reconstruction parameters are initialized with the average of the initial reconstruction values (filtering out outliers) if multiple camera pairs reconstructed the same scene plane. The global solution of this system will provide the final results, up to one free scale factor.

The proposed method was extensively evaluated on synthetic data in different configurations, and also proved good performance on multiple real data experiments, also comparing favorable to the State-of-the-Art general reconstruction method COLMAP [23] on the KITTI [9] dataset (see Fig. 5 for a comparative result).

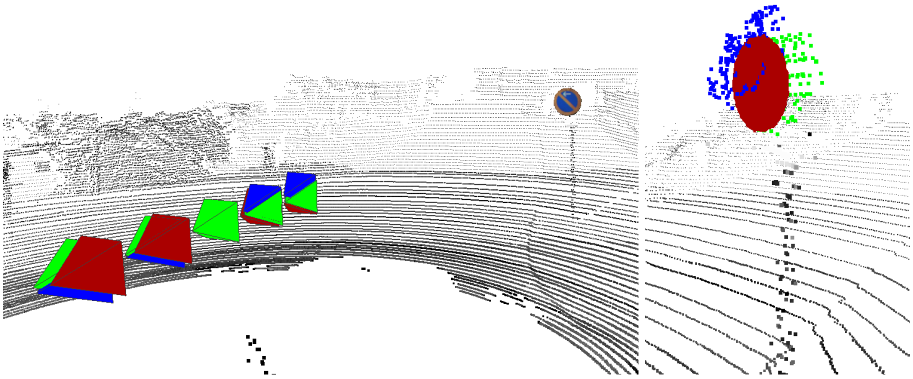

Figure 5. Comparative results of the propsed method and COLMAP[23]: Camera poses (left) and a traffic sign reconstruction (right) shown in green (ground truth), red (proposed), and blue (COLMAP).

Summary of the Author's Contributions

In the following, I summarized my results into two main thesis groups. In the first one, I present my findings on 2D-3D absolute pose estimation and visual data fusion, while in the second one my results on planar homography estimation and 3D reconstruction are shown. In Table A.1., the connections between the thesis points and the corresponding publications are displayed.

I. **Absolute Pose Estimation and Data Fusion**
Inspired by the 2D registration framework of [1], [2] proposed a novel formulation of the absolute pose estimation of a perspective camera with respect to a 3D depth data as a general 2D-3D registration that works without the use of any dedicated calibration pattern or explicit point correspondences. This idea can be extended into a general framework for the absolute pose estimation of central spherical cameras, and applied for different visual data fusion tasks. The basic idea is to set up a system of non-linear equations whose solution directly provides the parameters of the aligning transformation. This thesis group summarizes my results on the absolute pose estimation topic and two data fusion applications.

   (a) I experimentally tested the performance of the absolute pose estimation algorithm of omnidirectional cameras introduced in [Tamas, Frohlich, Kato, 2014] on synthetic data. For a common registration framework for central cameras I implemented the proposed spherical surface integral calculation that reformulates [Tamas, Frohlich, Kato, 2014] to work with triangles of a mesh representation, and I deducted an efficient 2D geometric moments calculation scheme for the surface integrals of perspective cameras presented in [2]. I proposed an initialization step of the rotation and translation parameters for both spherical and perspective cameras, that works automatically using the projection of the corresponding 2D-3D regions. Through quantitative evaluation of the method, I proved its performance, I compared it to previous point-wise spherical integral approximation approach [Tamas, Frohlich, Kato, 2014] on large scale synthetic data, while also comparing the spherical and classical models applied for the perspective camera. I also demonstrated the performance and usability of the method on multiple real data test cases with different cameras and 3D sensors.

   (b) For the first visual data fusion application for cultural heritage objects, I adapted our region-based registration method [Tamas, Frohlich, Kato, 2014] extending it to non-planar, smooth surfaces. As part of the workflow, I proposed an ICP refinement step based on intensity data edges, and a simple solution for the multi-camera fusion problem based on the cameras' orientation. I experimentally proved that despite the change to non-planar surfaces, the robustness of the method remains the same, while also conducting real tests on collected data of cultural heritage objects. The second application focuses on the selection of views from large number of cameras. I implemented a more complex camera selection algorithm, to fully benefit from the different focal length, resolution and position of cameras, based on multiple criteria, like visibility, sharpness, viewing angle and resolution. Visualizing the fusion results required a solution for the correct texture mapping between the 3D model and hundreds of texture image files, thus I proposed a technical solution that can easily use the original images as textures, without the need to create specially baked texture files. I validated the proposed pipeline on the acquired 2D-3D large scale dataset of two Reformed churches.

II. **Planar Homography Estimation and 3D Reconstruction**

The 2D registration framework of [1] can also be extended for estimating planar homographies between spherical cameras. Practically the homographies would act in this case between the spherical projections in the two cameras, representing the image of the same planar region. In general, relative pose parameters, as well as the normal and distance of the inducing plane can be factorized from such a planar homography, but due to the inherent parametrization of a planar homography, direct approaches for solving the problem are also possible, avoiding the factorization step completely. This thesis group summarizes my results on the planar homography estimation and 3D reconstruction topics.

(a) I experimentally validated the proposed region-based homography estimation method for omnidirectional cameras using two of the most commonly used models. Following [18] I deducted the decomposition of relative pose parameters from homographies assuming a weak Manhattan world constraint, then proved its comparable performance to the standard factorization method of [17] on synthetic data. If relative pose is available, one can also calculate the parameters of the inducing planar patch from the homography. I validated the proposed differential geometric approach for the computation of the normal vector, using the homographies estimated by our method [Frohlich, Tamas, Kato, 2016]. Through comparative evaluation on synthetic data, I proved, that the proposed method outperforms the classical method of [20], and it is robust against noise in the rotation and translation parameters.

(b) Taking a different approach on the homography estimation problem with perspective cameras, a standard parametrization of the homography was applied through the relative pose and plane parameters. Each camera pair and each available region pair defines a new homography, thus I deducted the homography equations in a multi-camera multi-region setup through the common pose and plane parameters, and validated the algorithm both in a minimal case setup, and various configurations of cameras and regions. For the multi-camera setup I built a bundle adjustment to simultaneously estimate all the unknown parameters of the system. I experimentally proved the method's performance on synthetic and on real data with precise Lidar pointcloud and marker based measurements as reference, and also on the KITTI benchmark dataset where it proved State-of-the-Art performance in comparison to the point-based multi-view reconstruction method of [23].

	I		II	
	a	b	a	b
[Tamas, Frohlich, Kato, 2014]	●			
[Frohlich, Tamas, Kato, 2019]	●			
[Frohlich *et al.*, 2016]		●		
[Frohlich *et al.*, 2018]		●		
[Frohlich, Tamas, Kato, 2016]			●	
[Molnár *et al.*, 2014]			●	
[Frohlich, Kato, 2018]				●

Table 1. The connection between the thesis points and publications.

Publications

Articles

[Frohlich, Tamas, Kato, 2019] R. Frohlich, L. Tamas, and Z. Kato. "Absolute Pose Estimation of Central Cameras Using Planar Regions". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019). accepted subject to minor revision, under review, pp. 1–16.

Book Chapters

[Frohlich *et al.*, 2018] R. Frohlich, S. Gubo, A. Lévai, and Z. Kato. "3D-2D Data Fusion in Cultural Heritage Applications". In: *Heritage Preservation: A Computational Approach*. Ed. by B. Chanda, S. Chaudhuri, and S. Chaudhury. Springer Singapore, 2018, pp. 111–130. ISBN: 978-981-10-7221-5. DOI: 10.1007/978-981-10-7221-5_6.

[Frohlich, Tamas, Kato, 2016] R. Frohlich, L. Tamas, and Z. Kato. "Handling Uncertainty and Networked Structure in Robot Control". In: vol. 42. Studies in Systems, Decision and Control. Chapter 6. Springer, Feb. 2016. Chap. Homography Estimation Between Omnidirectional Cameras Without Point Correspondences, pp. 129–151.

Conference Papers

[Frohlich, Kato, 2018] R. Frohlich and Z. Kato. "Simultaneous Multi-View Relative Pose Estimation and 3D Reconstruction from Planar Regions". In: *Proceedings of ACCV Workshop on Advanced Machine Vision for Real-life and Industrially Relevant Applications*. Ed. by G. Carneiro. Vol. 11367. Lecture Notes in Computer Science. Springer, Dec. 2018. ISBN: ISBN 978-3-030-21074-8. DOI: 10.1007/978-3-030-21074-8.

[Frohlich *et al.*, 2016] R. Frohlich, Z. Kato, A. Tremeau, L. Tamas, S. Shabo, and Y. Waksman. "Region Based Fusion of 3D and 2D Visual Data for Cultural Heritage Objects". In: *Proceedings of International Conference on Pattern Recognition*. IEEE. Cancun, Mexico: IEEE, Dec. 2016, pp. 2404–2409.

[Molnár *et al.*, 2014] J. Molnár, R. Frohlich, C. Dmitry, and Z. Kato. "3D Reconstruction of Planar Patches Seen by Omnidirectional Cameras". In: *Proceedings of International Conference on Digital Image Computing: Techniques and Applications*. Wollongong, Australia: IEEE, Nov. 2014, pp. 1–8. ISBN: ISBN 978-1-4799-5409-4.

[Tamas, Frohlich, Kato, 2014] L. Tamas, R. Frohlich, and Z. Kato. “Relative Pose Estimation and Fusion of Omnidirectional and Lidar Cameras”. In: *Proceedings of the ECCV Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving*. Ed. by L. de Agapito, M. M. Bronstein, and C. Rother. Vol. 8926. Lecture Notes in Computer Science. Zurich, Switzerland: Springer, Sept. 2014, pp. 640–651. ISBN: ISBN 978-3-319-16180-8.

References

[1] C. Domokos, J. Nemeth, and Z. Kato. “Nonlinear Shape Registration without Correspondences”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.5 (May 2012), pp. 943–958. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2011.200.

[2] L. Tamas and Z. Kato. “Targetless Calibration of a Lidar - Perspective Camera Pair”. In: *Proceedings of ICCV Workshop on Big Data in 3D Computer Vision*. IEEE. Sydney, Australia: IEEE, Dec. 2013, pp. 668–675.

[3] D. Scaramuzza, A. Martinelli, and R. Siegwart. “A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion”. In: *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems*. ICVS-06. Washington, USA: IEEE Computer Society, 2006, pp. 45–51.

[4] D. Scaramuzza, A. Martinelli, and R. Siegwart. “A Toolbox for Easily Calibrating Omnidirectional Cameras.” In: *IEEE/RSJ International Conference on Intelligent Robots*. Bejing: IEEE, Oct. 2006, pp. 5695–5701.

[5] D. Herrera C, J. Kannala, and J. Heikkila. “Joint Depth and Color Camera Calibration with Distortion Correction.” In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.10 (2012), pp. 1–8.

[6] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis. “3D LIDAR-camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization”. In: *The International Journal of Robotics Research* 31.4 (2012), pp. 452–467.

[7] J. E. G. Joseph O’Rourke, ed. *Handbook of Discrete and Computational Geometry, Second Edition (Discrete Mathematics and Its Applications)*. Chapman and Hall/CRC, 2004. ISBN: 9781584883012.

[8] J. M. Pozo, M. C. Villa-Uriol, and A. F. Frangi. “Efficient 3D Geometric and Zernike Moments Computation from Unstructured Surface Meshes”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.3 (Mar. 2011), pp. 471–484. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2010.139.

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. “Vision meets Robotics: The KITTI Dataset”. In: *International Journal of Robotics Research (IJRR)* (2013), pp. 1231–1237.

[10] Z. Taylor and J. Nieto. “A Mutual Information Approach to Automatic Calibration of Camera and Lidar in Natural Environments”. In: *Australian Conference on Robotics and Automation*. Wellington, Australia, Dec. 2012, pp. 3–5.

[11] P. J. Besl and N. D. McKay. “Method for registration of 3-D shapes”. In: *Robotics-DL tentative*. International Society for Optics and Photonics. 1992, pp. 586–606.

[12] S. Katz, A. Tal, and R. Basri. “Direct Visibility of Point Sets”. In: *ACM SIGGRAPH 2007 Papers*. SIGGRAPH ’07. San Diego, California: ACM, 2007. DOI: 10.1145/1275808.1276407.

[13] G. Yang and B. J. Nelson. “Wavelet Based Autofocusing and Unsupervised Segmentation of Microscopic Images”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2003, Las Vegas, Nevada, October*. Vol. 3. IEEE, 2003, 2143–2148 vol.3. DOI: 10.1109/IROS.2003.1249188.

[14] M. Garland and P. S. Heckbert. “Surface simplification using quadric error metrics”. In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques - SIGGRAPH ’97* (1997), pp. 209–216. ISSN: 00978930. DOI: 10.1145/258734.258849.

[15] K. McHenry and P. Bajcsy. *An overview of 3D data content, file formats and viewers*. Tech. rep. 2008, p. 21.

[16] C. Mei, S. Benhimane, E. Malis, and P. Rives. “Efficient Homography-Based Tracking and 3-D Reconstruction for Single-Viewpoint Sensors”. In: *Robotics, IEEE Transactions on* 24.6 (Dec. 2008), pp. 1352–1364. ISSN: 1552-3098. DOI: 10.1109/TRO.2008.2007941.

[17] O. Faugeras and F. Lustman. *Motion and structure from motion in a piecewise planar environment*. Tech. rep. RR-0856. June 1988.

[18] 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.

[19] P. Sturm. “Algorithms for plane-based pose estimation”. In: *Proceedings of International Conference on Computer Vision and Pattern Recognition*. Vol. 1. June 2000, pp. 706–711. DOI: 10.1109/CVPR.2000.855889.

[20] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge, UK: Cambridge University Press, 2004.

[21] J. Molnár and D. Chetverikov. “Quadratic Transformation for Planar Mapping of Implicit Surfaces”. In: *Journal of Mathematical Imaging and Vision* 48.1 (2014), pp. 176–184.

[22] O. Faugeras and F. Lustman. *Motion and structure from motion in a piecewise planar environment*. Tech. rep. RR-0856. Sophia Antipolis, France: INRIA, June 1988.

[23] J. L. Schonberger and J.-M. Frahm. “Structure-from-Motion Revisited”. In: *Proceedings of International Conference on Computer Vision and Pattern Recognition*. IEEE, June 2016. DOI: 10.1109/cvpr.2016.445.