Fast and Accurate Refinement Method for 3D Reconstruction from Stereo Spherical Images

Marek Solony[1], Evren Imre[2], Viorela Ila[1], Lukas Polok[1], Hansung Kim[2] and Pavel Zemcik[1]

[1]*Faculty of Information Technology, Brno University of Technology, Brno*

[2]*Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford*

{*isolony, ila, ipolok, zemcik*}*@fit.vutbr.cz,* {*h.imre, h.kim*}*@surrey.ac.uk*

Keywords: spherical images registration, 3D reconstruction, graph optimisation, SLAM.

Abstract: Realistic 3D models of the environment are beneficial in many fields, from natural or man-made structure inspection and volumetric analysis, to movie-making, in particular, special effects integration to natural scenes. Spherical cameras are becoming popular in environment modelling because they capture the full surrounding scene visible from the camera location as a consistent seamless image at once. In this paper, we propose a novel pipeline to obtain fast and accurate 3D reconstructions from spherical images. In order to have a better estimation of the structure, the system integrates a joint camera pose and structure refinement step. This strategy proves to be much faster, yet equally accurate, when compared to the conventional method, registration of a dense point cloud via iterative closest point (ICP). Both methods require an initial estimate for successful convergence. The initial positions of the 3D points are obtained from stereo processing of pair of spherical images with known baseline. The initial positions of the cameras are obtained from a robust wide-baseline matching procedure. The performance and accuracy of the 3D reconstruction pipeline is analysed through extensive tests on several indoor and outdoor datasets.

1 INTRODUCTION

Nowadays, there is a large interest in recreating the 3D environment from images coming either from still or moving cameras, in stereo or monocular setups. Well-known applications such as Photo Tourism can reconstruct tourist places from thousands of internet images; Google's 3D maps application can reconstruct entire cities from aerial and satellite images.

Due to the limited field of view of the conventional cameras a large number of images needs to be acquired in order to reconstruct outdoor or indoor scenes. In applications involving large scenes, an acceptable coverage with still images can be a problem due to both, time consuming acquisition process as well as large memory requirements. One solution to this is to use spherical cameras which cover 360° of the space. Only few such images are needed to create a dense reconstruction of a large scene.

In this paper, we introduce a novel spherical image processing pipeline to obtain fast and accurate 3D reconstructions of a scene. The advantage of this system is that it can recreate relatively large outdoor or indoor scenes from only a handful of spherical images. In order to obtain a dense 3D reconstruction, stereo pairs of spherical images are considered. Acquiring stereo pairs is in general a simple task, since most of the spherical imaging systems are placed on a tripod which allows an exact adjustment of the height of the camera, without introducing any rotations.

The processing pipeline has two parts; a) an initialisation step and b) a structure and camera pose refinement step. The initial positions of the 3D points are obtained from a stereo processing framework which produces an accurate disparity map from stereo spherical image pairs with known baseline. To obtain accurate depth from stereo pairs, dense disparity maps are estimated using a multi-resolution partial differential equation (PDE) (Kim and Hilton, 2013) based stereo matching algorithm which accelerates the calculation for large images whilst keeping smooth surface and sharp object boundaries. The initial positions of the cameras are obtained from a robust wide-baseline matching procedure. The wide-baseline nature of the problem necessitates the use of robust estimators, such as RANSAC. Our geometry estimation pipeline is an implementation of the guided matching algorithm of (Hartley and Zisserman, 2003). It iterates over successive stages of matching and RANSAC-based geometry estimation.

The initialisation is further used to either register consecutive dense point clouds using Iterative Closest Point (ICP) (Kim and Hilton, 2013) algorithms or to perform joint pose and structure refinement. The ICP algorithm is widely used for geometric registration of three dimensional point clouds when an initial estimate of the relative pose is known (Besl and McKay, 1992; Rusinkiewicz and Levoy, 2001). ICP simply finds a 3D rigid transformation matrix to minimize the distance of the closest points between two point clouds. It suffers from local minimum problem if the initialisation is not close enough to the true solution.

On the other hand, the nonlinear structure refinement is based on joint optimisation of sparse 3D points and camera poses. The problem is formulated as a nonlinear optimisation on graphs (Dellaert and Kaess, 2006; Kümmerle et al., 2011), where the nodes are the 3D points and the 3D camera poses and the edges are the relative point-camera transformations (see section 4). The optimisation problem finds the best cameras-points configuration, given the imprecise relative positions of the 3D points obtained from the initialisation step. Section 5 analyses the time and performance of both approaches, the ICP based registration and the proposed refinement method.

2 Related Work

Building a representation of the environment from noisy sensorial data is a central problem in computer vision and robotics. Problems in computer vision include bundle adjustment (BA) (Agarwal et al., 2009) and structure from motion (SFM) (Beall et al., 2010). Simultaneous localization and mapping (SLAM) (Dellaert and Kaess, 2006; Kaess et al., 2008; Kümmerle et al., 2011) is a similar problem in robotics. Those are mathematically equivalent techniques, with a slight difference in the types of constraints: while BA minimizes the *reprojection error*, SLAM minimizes the residual directly in the representation space. The key to developing fast methods in this direction is the interpretation of the problem in terms of graphical models. Understanding the existing graphical model inference algorithms and their connection to matrix factorization methods from linear algebra allowed computationally efficient solutions (Dellaert and Kaess, 2006; Snavely et al., 2006; Kümmerle et al., 2011; Polok et al., 2013a).

For the particular problem of 3D reconstruction from stereo spherical images, minimizing the reprojection error in the image space may not be the best approach. This is due to the fact that in the spherical images, the reprojection error, which is the distance between measured and estimated 2*D* point, varies with the latitude of unwrapped spherical image. Therefore, a more correct approach is to optimize directly in the representation space applying SLAM optimisation methods.

While the BA approaches use sets of still images from unrelated viewpoints and possibly different cameras with different parameters, SfM uses video sequences from a moving camera to reconstruct a 3D model of the observed environment. The disadvantage of both, is the great number of images needed to cover large areas. A common way to capture the full 3D space is to use a catadioptric omnidirectional camera using a mirror combined with a charge-coupled device (CCD) (Hong et al., 1991; Nayar, 1997). Lhuillier proposed a scene reconstruction system using omnidirectional images (Lhuillier, 2008). However, catadioptric omnidirectional cameras have a large number of systematic parameters to calibrate. Another problem is the limited resolution because they use only one CCD to capture the full 3D space. Feldman and Weinshall used a Cross Slits (X-Slits) projection with a rotating fisheye camera to acquire high quality spherical images while reducing the number of camera parameters and to generate distortion-free image-based rendering (Feldman and Weinshall, 2005). Kim and Hilton used this spherical imaging to acquire multiple pairs of high-resolution stereo spherical images and reconstructed full 3D geometry of the scene (Kim and Hilton, 2013). Note that using omnidirectional or wide field of view cameras decreases the ratio of the number of cameras to the number of observed points, as opposed to the traditional cameras. As will be shown later, this is beneficial for efficient optimisation, as the resulting system matrix sparsity patterns are favourable for solving using Schur complement (Zhang, 2005).

3 Spherical Image Processing

Using stereo spherical images from one or multiple view-points is an easy and feasible way to create 3D models of large environments. To obtain a 3D structure from stereo spherical images, image processing algorithms are applied followed by a structure refining procedure. This section describes an image processing pipeline which inputs spherical images and outputs an initial estimation for the refining step.

3.1 Spherical Images

A spherical image is captured by a vertical line-scan camera with wide-angle lens rotating around the centre of projection. The final image is created by joining scans into a single image so it covers 360° in horizontal and $\approx 180^\circ$ in vertical field of view. This process is equivalent to projecting the scene around the camera onto a unit sphere and unwrapping it into a plane.

Compared to the images from conventional cameras, the spherical images contain more information in a single image and therefore more features can be extracted for the purpose of estimating the relations between cameras. The main disadvantage of the spherical images is the distortion introduced by the wide-angle lenses and the rotating camera sensor. The same parts of the scene viewed from different positions of the camera, appear very different (see Fig. 1 b)). This can cause problems when extracting and matching features, especially when the images are captured with wide baseline.

3.2 Stereo Processing

A stereo image pair with vertical disparity can be obtained by placing the spherical camera at different heights. Assuming that the images are precisely aligned, the stereo-matching problem can be reduced to a one-dimensional search on a line. The disparity map is computed by processing all the columns of the stereo image pair, and therefore the 3D position of any valid 2D point can be obtained through a simple triangulation. If pixels on the column are mapped to the $[0, \pi]$ range in the spherical coordinate, the disparity d between projection points $p_t(x_t, y_t)$ and $p_b(x_b, y_b)$ of a 3D point P to the stereo image pair I_t and I_b is defined as the difference of the vertical angles of the projected points θ_t and θ_b as in:

$$d(p_t) = \theta_t - \theta_b \,. \tag{1}$$

The depth D_t (the distance between the top camera and the 3D point P) can be calculated by triangulation as in Eq. (2) using the baseline distance B in addition to the angular disparity.

$$D_t(p_t) = B / \left(\frac{\sin \theta_t}{\tan(\theta_t + d(p_t))} - \cos \theta_t \right) \tag{2}$$

A number of studies have been reported on the disparity estimation problem since the 1970 (Scharstein and Szeliski, 2002). Any disparity estimation algorithm can be used for our system as long as it produces accurate and dense disparity. Most disparity estimation algorithms solve the correspondence problem on a discrete domain such as integer or half-pixel levels which are not sufficient to recover a smooth surface. Especially spherical stereo image pairs can show more serious artefacts in the reconstruction because they have a serious radial distortion. A variational approach which theoretically works on a continuous domain can be a solution for accurate floating-point disparity estimation. We use a PDE-based variational disparity estimation method to generate accurate disparity fields with sharp depth discontinuities for surface reconstruction (Kim and Hilton, 2013).

3.3 Feature and Descriptors Extraction

The estimation of the relative pose between two spherical cameras requires a set of reliable 3D point correspondences. 3D features from each camera are selected by detecting 2D SIFT features on the spherical images, and computing the associated 3D coordinates, by projecting them to the 3D space via the computed depth map (section 3.2). Each 3D point is characterised by the 2D SIFT descriptor of its corresponding 2D feature on the spherical image (İmre et al., 2010). This process generates a sparse 3D point cloud for each camera and its corresponding descriptor as input to the initial pose estimation stage.

3.4 Computation of the Initial Pose Estimate

The computation of the initial pose estimate follows the conventional guided matching pipeline, which involves alternating stages of feature matching and geometry estimation (Hartley and Zisserman, 2003). The pipeline requires two sets of 3D features as input. It returns $\mathcal{H}_i$, the initial 3D pose estimate, and I, the 3D correspondences supporting $\mathcal{H}_i$.

The feature matching stage seeks for nearest neighbours, by comparing the associated SIFT descriptors (Lowe, 2004). However, the pipeline often operates under wide-baseline conditions, which significantly reduces the number of viable matchings. Therefore, the implementation resorts to a compromise between ambiguity and quantity, and considers the 7 nearest neighbours, instead of the best. Each candidate is verified for *reciprocity*, *i.e.* whether the points are in each other's neighbourhoods. Excessively ambiguous matches are rejected by truncating the neighbourhoods so that, the ratio of the similarity scores for the worst candidate within the neighbourhood and the best candidate without is above a threshold. The remaining correspondences are ranked by the MR-Rayleigh metric (V.Fragoso and Turk, 2013).

The wide-baseline nature of the problem and the multiple-element neighbourhoods imply a correspondence set with many outliers. In geometry estimation, such problems are typically solved by the help of RANSAC (Fischler and Bolles, 1981). RANSAC applies a hypothesise-and-test framework on small, randomly selected sets of correspondences, in its search for a set without any outliers.

For pose estimation, we generate the hypotheses via (Horn, 1987), which requires 3-element samples. Our RANSAC implementation minimises the symmetric transfer error (Hartley and Zisserman, 2003). It makes use of MSAC (Torr and Zisserman, 2000), LO-RANSAC (Chum et al., 2003), biased sampling (Chum and Matas, 2005) and WaldSAC (Chum and Matas, 2008). MSAC and LO-RANSAC improves the estimate accuracy, through better hypothesis assessment, and a local optimisation step to improve promising hypotheses, respectively. Biased sampling steers the hypothesis generation towards samples with a better likelihood of being inliers (as indicated by the correspondence ranking). WaldSAC allows the rejection of poor hypotheses without testing the entire correspondence set, and therefore, provides significant computational savings. RANSAC terminates when it is confident that a better solution is unlikely (Chum and Matas, 2008), returning $\mathcal{H}_i$ and I.

3.5 Iterative Closest Point for 3D Point Cloud Registration

One way to obtain a 3D dense reconstruction from spherical stereo images is to use the camera pose estimation from the previous step as an initialisation of a dense 3D point-cloud registration. The 3D points are obtained from the depth map computed as in 3.2 and they are registered using ICP algorithm.

The ICP algorithm has been widely adopted to align two given point sets (Besl and McKay, 1992; Rusinkiewicz and Levoy, 2001). It finds a rigid 3D transformation (rotation R and translation t) between two overlapping clouds of points by iteratively minimising squared-error of registration between the nearest points from one set to the other:

$$E_R(R,t) = \sum_{i}^{N_m}\sum_{j}^{N_d} w_{i,j}\|m_i - (Rd_j + t)\|^2 \qquad (3)$$

where N_m and N_d are the number of points in the model set m and reference set d, respectively, and $w_{i,j}$ are the weights for a point match.

In each ICP iteration, the rigid 3D transformation can be efficiently calculated by either singular value decomposition (SVD) (Arun et al., 1987) or the closed-form solution (Horn, 1987). The closed-form solution solves the least-squares problem for three or more points using a unit quaternion to represent rotation or using manipulation matrices and their eigenvalues-eigenvector decomposition.

When applied to dense point cloud registration, the ICP algorithm can become very slow. Therefore, in this work, we propose a method which refines a sparse structure and the camera poses, and creates the dense reconstruction by referring the dense point-clouds to the accurately estimated camera positions.

4 Joint Pose and Structure Refinement

In order to obtain an accurate 3D structure, the proposed pipeline performs a joint optimisation of the initial camera poses and the corresponding sparse point clouds. The problem is formulated as a nonlinear optimisation on graphs, where the vertices are the absolute points and camera poses and the edges are relative point-camera transformations obtained from the depth map. Said differently, the vertices are the variables to be estimated and the edges are the measured constraints. To obtain an optimal configuration of the graph, we perform a maximum likelihood estimation (MLE).

Under the assumption of zero-mean Gaussian measurement noise, the MLE has a nonlinear least squares (NLS) solution. The goal is to obtain the MLE of a set of variables $\theta = [\theta_1 \dots \theta_n]$, containing the 3D points in the environment $\mathbf{p} = [p_1 \dots p_{np}]$ and the camera poses $\mathbf{c} = [c_1 \dots c_{nc}]$, given the set of 3D relative measurements, $\mathbf{z} = [z_1 \dots z_m]$:

$$\theta^* = \operatorname*{argmax}_{\theta} P(\theta \mid \mathbf{z}) = \operatorname*{argmin}_{\theta} \{-\log(P(\theta \mid \mathbf{z})\} \quad (4)$$

This measurement can be modelled as a function $h_k(c_i, p_j)$ of the camera c_i and the point p_j with zero-mean Gaussian noise with the covariance Σ_k:

$$P(z_k \mid c_i, p_j) \propto \exp\left(-\frac{1}{2} \parallel z_k - h(c_i, p_j) \parallel^2_{\Sigma_k}\right), \quad (5)$$

Finding the MLE from (4) is done by solving the following nonlinear least squares problem:

$$\theta^* = \operatorname*{argmin}_{\theta} \left\{ \frac{1}{2} \sum_{k=1}^{m} \left\| z_k - h(c_i, p_j) \right\|^2_{\Sigma_k} \right\}. \quad (6)$$

Iterative methods such as Gauss-Newton (GN) or Levenberg-Marquard (LM) are used to find the solution of the NLS in (6). An iterative solver starts with an initial point θ^0 and, at each step, computes a correction δ towards the solution. For small $\|\delta\|$, a Taylor series expansion leads to linear approximations in the neighbourhood of θ^0:

$$\tilde{\mathbf{e}}(\theta^0 + \delta) \approx \mathbf{e}(\theta^0) + J\delta, \quad (7)$$

where $\mathbf{e} = [e_1, \dots, e_m]^\top$ is the set of all nonlinear errors between the estimated and the actual measurement, $e_k(c_i, p_j, z_k) = z_k - h(c_i, p_j)$, with $[c_i, p_j] \subseteq \theta$

and J is the Jacobian matrix which gathers the derivative of the components of $\mathbf{e}$. Thus, at each iteration i, a linear LS problem is solved:

$$\delta^* = \operatorname*{argmin}_{\delta} \frac{1}{2} \|A\,\delta - \mathbf{b}\|^2 \,, \tag{8}$$

where the $A = \Sigma^{-\top\backslash 2} J(\theta^i)$ is the system matrix, $\mathbf{b} = \mathbf{e}(\theta^i)$ the right hand side (r.h.s.) and $\delta = (\theta - \theta^i)$ the correction to be calculated (Dellaert and Kaess, 2006). The the minimum is attained where the first derivative cancels:

$$A^\top A\,\delta = A^\top \mathbf{b} \quad \text{or} \quad \Lambda\delta = \eta \,, \tag{9}$$

with $\Lambda = A^\top A$, the square symmetric system matrix and $\eta = A^\top \mathbf{b}$, the right hand side. The solution to the linear system can be obtained either by sparse matrix factorization followed by backsubstitution or by linear iterative methods. After computing δ, the new linearisation point becomes $\theta^{i+1} = \theta^i \oplus \delta$.

In our application the initial solution θ^0 can be relatively far from the minimum, therefore LM is preferred over the GN methods. LM is based on efficient damping strategies which allow convergence even from poor initial solutions. For that, LM solves a slightly modified variant of (9), which involves a damping factor λ:

$$(\Lambda + \lambda\bar{D})\delta = \eta \quad \text{or} \quad H\delta = \eta \,, \tag{10}$$

where $\bar{D}$ can be either the identity matrix, $\bar{D} = I$, or the diagonal of the matrix Λ, $\bar{D} = \operatorname{diag}(\Lambda)$.

Schur complement is employed to solve the linearised problem in (10). For that, the system matrix is split in four blocks, according to the camera and points variables:

$$\begin{bmatrix} C & U \\ U^\top & P \end{bmatrix} \cdot \begin{bmatrix} \mathbf{c} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \eta_c \\ \eta_p \end{bmatrix} \tag{11}$$

This is a common practice in solving 3D reconstruction problems, where the camera poses are linked only through the points. It results in diagonal C and P matrices, which can be easily inverted. If P is invertible, the Schur complement of the block P is $C - UP^{-1}U^\top$, and is used to solve for the camera poses first. Points are then obtained by solving the remaining system.

Performing matrix inversion and multiplication in the Schur complement form brings reduction in computational time, compared to performing Cholesky factorisation of the whole system. To improve convergence, after every iteration of the nonlinear solver, the state of the cameras is fixed and three iterations optimising only the points are performed. This is based on the observation, that while a single camera may depend on a large number of points, a single point is usually only observed from two or three cameras, and as such the positions of the points are harder to estimate precisely. The extra iterations allow the points to settle before performing the following nonlinear solver iterations. In our experiments, the extra iterations reduce residual as efficiently as the full nonlinear iterations, only at much smaller computational cost. A similar technique was described in (Jeong et al., 2012).

The result of the estimation is an optimised configuration of the sparse 3D structure $\mathbf{p}$ and the camera poses $\mathbf{c}$. The dense structure is obtained by referring, for each pixel, its 3D position from the depth map to the optimised camera poses.

5 Evaluations and Results

This section evaluates the automatic process of aligning multiple spherical cameras in the terms of accuracy and computation time.

5.1 Datasets

Multiple experiments were performed in order to evaluate the performance of the proposed 3D reconstruction from stereo spherical images using three spherical image datasets; two outdoor datasets, covering large area: Cathedral and CCSR, and one indoor dataset with measured ground truth: Studio. The images were acquired with a SpheroCam-HDR system, which captures scan lines with turning fisheye lenses, synthesises them and provides a 50 Mpix latitude-longitude image. For Cathedral and CCSR dataset, the camera was placed in three different positions around the scene and for Studio dataset, four positions have been used. Each capture was done at two different heights to produce stereo image pairs.

The Cathedral dataset covers an area of about 2500 m^2. In order to test how the system performs in the case of large sensor displacements, the cameras were placed at positions far apart (approx. 18 m in the Cathedral dataset). The CCSR dataset has shorter sensor displacements, and covers an area about 250 m^2. Studio dataset was captured with the purpose to evaluate the pipeline in a short sensor displacement setting (approx. 1 m). All details about datasets are included in Table 1.

5.2 Evaluation modules

Several modules can be distinguished in the stereo image processing pipeline: depth map calculation, feature extraction, matching and initial pose and structure estimation and refinement. The steps are illustrated in Figure 1. The stereo spherical image pro-

Table 1: Top: Dataset descriptions; Bottom: Time required to process image pairs.

			
Characteristics	Studio	Cathedral	CCSR
Ground truth	Measured GT	ICP GT	ICP GT
Stereo baseline [m]	0.2	0.6	0.2
Sensor displacement [m]	1.00/2.00/1.00	19.38/17.04	5.81/5.22
Cameras	4	3	3
Matches	754/340/997	546/488	2584/2150
Vertices	1715	681	2832
Edges	3653	1425	6048
Processing			
Feat. & desc. extract [s]	8.15	7.02	6.32
Initial estimation [s]	6.99	11.41	25.65
Refinement			
ICP [s]	146.057	366.024	995.43
SLAM++[s]	**0.120**	**0.091**	**0.134**

cessing pipeline and the graph optimisation for pose and sparse structure refinement were implemented in C / C++. For the ICP registration a standard implementation provided by the PCL library (Rusu and Cousins, 2011) was used.

Iterative closest point (ICP)

Dense registration using ICP has been successfully used in the literature for the 3D reconstruction from spherical images (Kim and Hilton, 2013). Therefore, ICP is used as a reference in the time and accuracy evaluations of the proposed refinement method. To calculate the initial estimate of the camera poses and the 3D structure, the proposed image processing pipeline is used. In the following text, we refer to this simply as ICP.

Furthermore, we use dense ICP to define a ground truth for testing the accuracy of our method in the outdoor datasets where there are no manual measurements available. For that manually matched sparse features are used to calculate an initial estimate for the ICP registration, and it will be further refers in the paper as GT-ICP.

Joint pose and structure refinement

The joint pose and structure refinement is based on our open-source, nonlinear graph optimisation library, called SLAM++ (Polok et al., 2013b). This is a very efficient implementation of nonlinear least squares solvers, based on fast sparse block matrix manipulation for solving the linearised problems. SLAM++ produces fast, but accurate estimations, which many times outperform similar state-of-the-art implementations of graph optimisation systems. This performance it is achieved due to the fact that the implementation exploits the block structure the problems and offers very fast solutions to manipulate block matrices and perform arithmetic operations. Edges and vertices are defined according to section 4 and the optimisation is performed on the manifold, therefore it is correctly handling the derivatives of rotations and translations. There are no simplifications or approximations involved in the optimisation process.

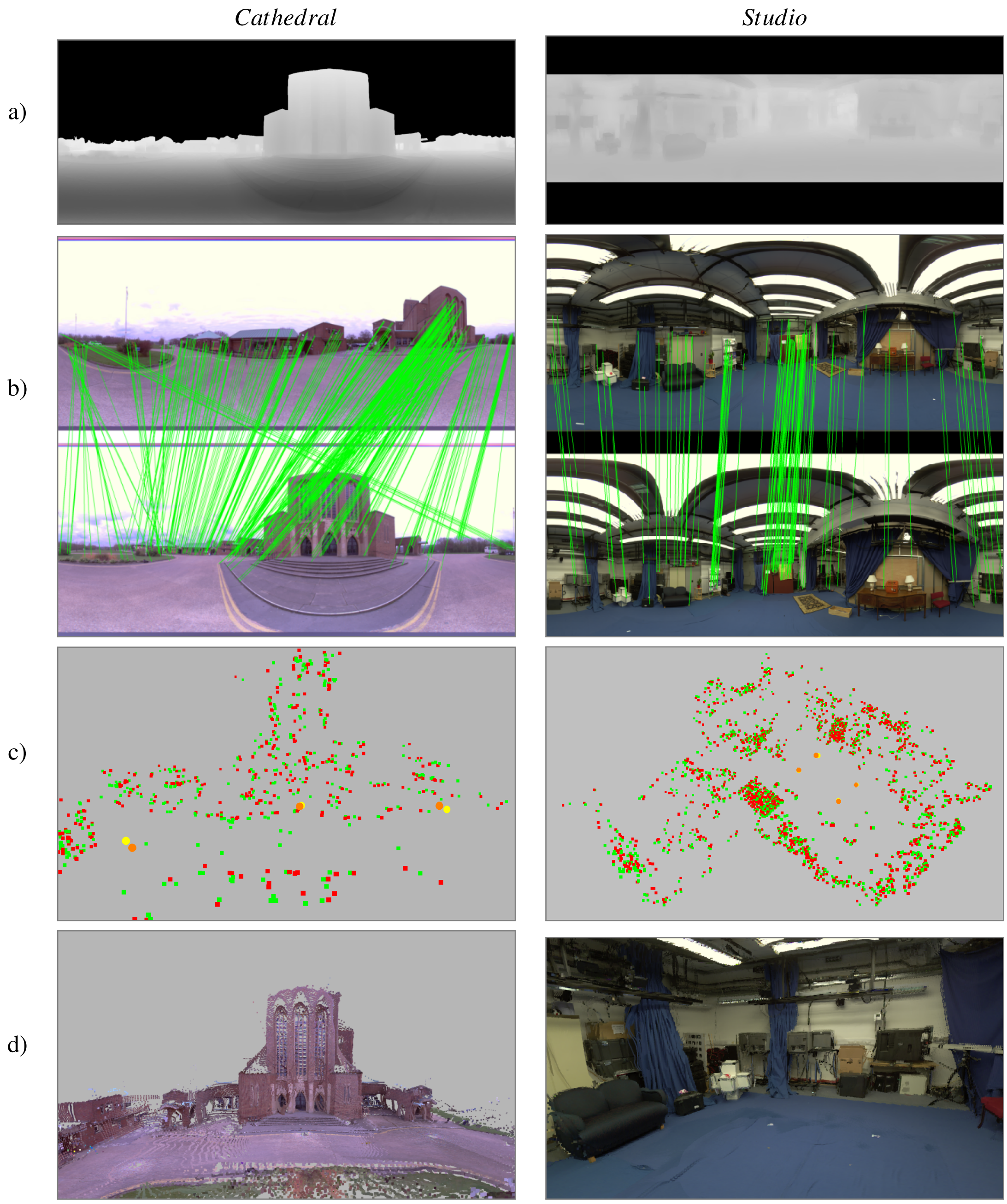
Table 2: Depth map accuracy results: Differences between GT and measurements in the depth map.

	Sofa 1	Sofa 2	Table	Carpet
P1 [cm]	1.8	1.2	0.2	0.3
P2 [cm]	-	-	0.7	0.8
P3 [cm]	-	-	7.8	0.1
P4 [cm]	1.7	3.2	1.3	0.3

5.3 Time evaluation

The disadvantage of applying ICP for image registration is its processing time. The approach proposed in this paper offer much faster solutions in this direction. Table 1, bottom, shows that SLAM++ is, for all datasets, almost three orders of magnitude faster than the ICP algorithm. The good time performance stems from the fact that it optimises for a sparse set of points and from the actual implementation based on sparse block solutions to non-linear least squares estimators.

Figure 1: Processing pipeline a) depth map b) inliers after matching with RANSAC algorithm (for better visibility only a fraction of matches is shown for Studio dataset). Please note that the crossing lines in the left column are not outliers, the image is spherical so the left part of the image continues on the right. c) initial (in red) vs. optimised (in green) poses and sparse point cloud d) final dense 3D reconstruction.



By analysing the processing time of each step of the pipeline in Table 1, we see that the time of optimising the camera poses is now very small compared to the other processing times in the pipeline, while using ICP, the registration time would have been the predominant time and would have constituted a bottleneck in large applications.

5.4 Accuracy evaluation

In our pipeline we can identify two sources of errors that can affect the final reconstruction, a) the error of the depth map and b) the camera pose estimation error. To analyse the accuracy of the proposed technique, ground truth data were measured for

Table 3: Accuracy results: Top: Structure Error. Bottom: Camera pose error evaluated separately for the rotation and translation.

Criteria	Method	Studio			Cathedral		CCSR	
		S1-S2	S2-S3	S3-S4	S1-S2	S2-S3	S1-S2	S2-S3
Pose Error	SLAM++ [cm]	0.46	0.79	1.12	70.87	37.14	37.49	11.93
	ICP [cm]	1.07	3.61	5.08	67.83	74.05	26.11	14.99
	SALM++ [degree]	1.14	0.57	0.89	5.48	3.91	0.81	1.66
	ICP [degree]	5.03	0.81	1.38	4.85	4.83	0.52	2.71
Structure Error	SALM++ [cm]	1.61			112.02		48.89	
	ICP [cm]	3.54			176.56		39.47	

all three datasets. Smaller sensor displacement and flat ground surface of the Studio dataset allowed for precise positioning of spherical cameras, and manual measurements of distances from the spherical camera positions to several objects in the scene as well as distances between camera poses. For the outdoor datasets, Cathedral and CCSR, the ground truth data was generated by manually matching sparse features to create an initialisation for the dense ICP (GT-ICP).

The error of the depth map was evaluated for the Studio dataset by comparing the calculated depth from the dense stereo processing with the measured ground truth. In this dataset, the cameras were placed in four different position with a know distance in between, and distance to objects in the scene (two distances to the sofa, one to the carpet, and one to the desk) were measured. Table 2 shows the errors in cm between the manually measured and the estimated 3D positions. The depth map error is, in average, of 1.6 cm for the Studio dataset. We can say that is a very good depth calculation from stereo spherical images for indoor scenes, nevertheless, we should expect larger errors in the outdoor scenes.

In order to evaluate the joint camera and structure estimation, two types of errors are evaluated, a) camera pose estimation error and b) structure error. To compute the pose estimation error, the transformations between the GT-ICP and the estimated poses are calculated. The errors in translation and rotation are reported separately, by computing the norm of the translation and the angle of rotation, respectively. For each dataset, pair-wise spherical camera registrations are evaluated. The structure error is computed in Studio dataset as average error of distances to known objects in scene and in the case of Cathedral and CCSR datasets the structure error is given by the average euclidean distance between two *dense* point clouds - one from GT-ICP and second from optimized solution.

Table 3 confirms our expectations that both, ICP and SLAM++ have similar accuracy, and that larger errors in pose estimation correlate with errors in structure estimation. Note that for longer baselines, the SLAM++ copes better with the errors in the initial estimation compared to ICP which requires very good initialisations. This is due to the fact that unlike ICP which relies only on matches between consecutive spherical cameras for each registration, SLAM++ also considers matches over multiple spherical images.

6 Conclusions and future work

The contribution of this paper is the formulation of the 3D reconstruction from spherical images in terms of sparse SLAM and based on that, obtaining a much faster, yet accurate solution than the existing methods based on dense ICP. The efficiency comes from both, the algorithm and the highly efficient sparse block matrix implementation of the nonlinear solver used in jointly refining the structure and the camera poses. An initial estimate of the 3D points and the camera positions is obtained from stereo processing of pair of spherical images with known baseline and a robust wide-baseline matching procedure. After the initialisation, the structure can be refined either via dense point cloud registration or joint camera pose and sparse structure optimisation. The later offers a much faster alternative to ICP while maintaining similar accuracy. The speed of the proposed approach was at least three orders of magnitude faster than ICP on all the datasets. It also offers a more robust estimation capable of exploiting relationships between multiple spherical images and refines the solution according to those constraints, whereas the ICP algorithm works only pair-wise. The same approach can be also easily applied in reconstruction from RGB-D cameras where the 2D image features and the corresponding 3D points can be refined similarly to those from stereo spherical images.

The proposed approach performs the optimisation on sparse structure and then transforms the dense point clouds by the calculated camera transformations. This yields a valid result, however, it may be possible to obtain a more precise dense point cloud alignment. Since the relation between the sparse points and points from the dense point cloud are known, it is possible to calculate a rigid transformation that aligns the dense point cloud to the corresponding sparse points (e.g. using (Horn, 1987)). Further work will expand in this direction.

7 Acknowledgements

The research leading to these results has received funding from the European Union, 7^{th} Framework Programme grants 316564-IMPART and the IT4Innovations Centre of Excellence, grant n. CZ.1.05/1.1.00/02.0070, supported by RDIOP funded by Structural Funds of the EU and the state budget of the Czech Republic.

REFERENCES

Agarwal, S., Snavely, N., Simon, I., Seitz, S. M., and Szeliski, R. (2009). Building rome in a day. In *Twelfth IEEE International Conference on Computer Vision (ICCV 2009)*, Kyoto, Japan.

Arun, K., Huang, T., and Blostein, S. (1987). Least square fitting of two 3-d point sets. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(5):698–700.

Beall, C., Lawrence, B., Ila, V., and Dellaert, F. (2010). 3D Reconstruction of Underwater Structures. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.

Besl, P. and McKay, N. (1992). A method for registration of 3-d shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):239–256.

Chum, O. and Matas, J. (2005). Matching with PROSAC - Progressive Sample Consensus. In *Proc. CVPR*, pages 220–226.

Chum, O. and Matas, J. (2008). Optimal Randomized RANSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1472–1482.

Chum, O., Matas, J., and Kittler, J. (2003). Locally Optimized RANSAC. In *Lecture Notes in Computer Science*, volume 2781, pages 236–243. Springer.

Dellaert, F. and Kaess, M. (2006). Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. of Robotics Research*, 25(12):1181–1203.

Feldman, D. and Weinshall, D. (2005). Realtime ibr with omnidirectional crossed-slits projection. In *Proc. ICCV*, pages 839–845.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395.

Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. 2nd edition.

Hong, J., Tan, X., Pinette, B., Weiss, R., and E.M., R. (1991). Image-based homing. In *Proc. ICRA*, pages 620–625.

Horn, B. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642.

İmre, E., Guillemaut, J.-Y., and Hilton, A. (2010). Moving Camera Registration for Multiple Camera Setups in Dynamic Scenes. In *Proc. BMVC*, pages 1–12.

Jeong, Y., Nister, D., Steedly, D., Szeliski, R., and Kweon, I.-S. (2012). Pushing the envelope of modern methods for bundle adjustment. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(8):1605–1617.

Kaess, M., Ranganathan, A., and Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365–1378.

Kim, H. and Hilton, A. (2013). 3d scene reconstruction from multiple spherical stereo pairs. *International Journal of Computer Vision*, 104(1):94–116.

Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., and Burgard, W. (2011). g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China.

Lhuillier, M. (2008). Automatic scene structure and camera motion using a catadioptric system. *Computer Vision and Image Understanding*, 109(2):186–203.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Nayar, S. (1997). Catadioptric omnidirectional camera. In *Proc. CVPR*, pages 482–488.

Polok, L., Ila, V., Solony, M., Smrz, P., and Zemcik, P. (2013a). Incremental block cholesky factorization for nonlinear least squares in robotics. In *Proceedings of the Robotics: Science and Systems 2013*.

Polok, L., Solony, M., Ila, V., Zemcik, P., and Smrz, P. (2013b). Efficient implementation for block matrix operations for nonlinear least squares problems in robotic applications. In *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE.

Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the ICP algorithm. In *Proc. 3DIM*, pages 146–152.

Rusu, R. B. and Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). In *Proc. ICRA*.

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42.

Snavely, N., Seitz, S. M., and Szeliski, R. (2006). Photo tourism: exploring photo collections in 3d. *ACM transactions on graphics (TOG)*, 25(3):835–846.

Torr, P. H. S. and Zisserman, A. (2000). MLESAC: A New Robust Estimator with Application to Estimating Image Geometry. *Computer Vision and Image Understanding*, 78(1):138–156.

V.Fragoso and Turk, M. (2013). SWIGS: A Swift Guided Sampling Method. In *Proc. of IEEE Conf. on Com-*

puter Vision and Pattern Recognition (CVPR), pages 2770–2777, Portland, Oregon.

Zhang, F. (2005). *The Schur complement and its applications*, volume 4. Springer.