SICE Journal of Control, Measurement, and System Integration, Vol. 10, No. 5, pp. 476–485, September 2017

Optical Flow-Based Epipolar Estimation of Spherical Image Pairs for 3D Reconstruction

Sarthak PATHAK *, Alessandro MORO *, Atsushi YAMASHITA *, and Hajime ASAMA *

Abstract: Stereo vision is a well-known technique for vision-based 3D reconstruction of environments. Recently developed spherical cameras can be used to extend the concept to all 360° and provide LIDAR-like 360 degree 3D data with color information. In order to perform accurate stereo disparity estimation, the accurate relative pose between the two cameras, represented by the five degree of freedom epipolar geometry, needs to be known. However, it is always tedious to mechanically align and/or calibrate such systems. We propose a technique to recover the complete five degree of freedom parameters of the epipolar geometry in a single minimization with a dense approach involving all the individual pixel displacements (optical flow) between two camera views. Taking advantage of the spherical image geometry, a non-linear least squares optimization based on the dense optical flow directly minimizes the angles between pixel displacements and epipolar curves in order to align them. This approach is particularly suitable for dense 3D reconstruction as the pixel-to-pixel disparity between the two images can be calculated accurately and converted to a dense point cloud. Further, there are no assumptions about the direction of camera displacement. We demonstrate this method by showing some error evaluations, examples of successfully rectified spherical stereo pairs, and the dense 3D models generated from them.

Key Words: stereo vision, spherical camera, 3D reconstruction.

1. Introduction

Stereo vision is a powerful technique for 3D reconstruction of environments. It usually makes use of two or three cameras and attempts to find the disparity of image points from one image to the other, which can then be transformed to 3D positions using the intrinsic parameters of the camera. The process of finding the disparity can be made much easier if the accurate five degree of freedom epipolar geometry describing the relative positions of the two cameras is known [1]. The images can be 'rectified' to bring the pixels of corresponding points to the same horizontal line thereby reducing it to a 1-dimensional search. Usually, this is done using checkerboards/special patterns or by mechanical alignment of the two cameras.

Recently, spherical cameras such as the Ricoh Theta (Fig. 1) have become popular. Their epipolar geometry is vastly different from that of normal cameras and has been studied before [2]. In spherical cameras, corresponding points follow epipolar circles, instead of epipolar lines, as shown in Fig. 2. Spherical cameras combined with stereo vision provide a powerful tool for 3D reconstruction in all 360°. Hence, many researchers have attempted the use of spherical imaging systems for 3D reconstruction ([3]–[5]). However, they all used some special patterns (checkerboards, cubic checkerboards, etc.) or relied on mechanical alignment for estimating or fixing the epipolar geometry. Such methods are tedious and introduce restrictions in camera alignment. Moreover, they are not possible unless there are two separate cameras fixed on a rig. One of the main advantages of using spherical stereo cameras, i.e., to allow camera alignment in any desired configuration, is lost.

(a) (b)

Fig. 1 An example of a complete spherical image in (a) equirectangular projection (b) spherical projection. They can be rotated in any direction without information loss.

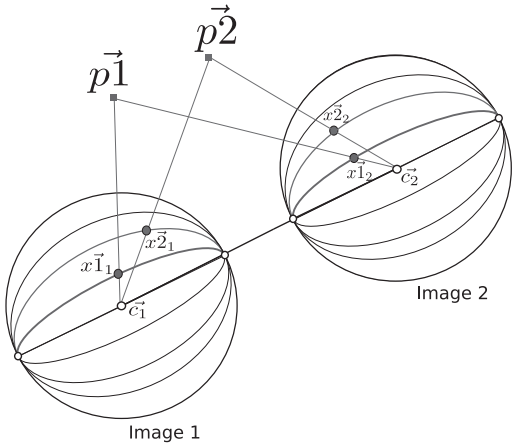


Fig. 2 In the spherical camera model, corresponding points follow a circular path, as opposed to straight lines in regular cameras. $\vec{p1}$ and $\vec{p2}$ are points in the real world. $x\vec{1}_i$ and $x\vec{2}_i$ are their projections in Image i. It can be shown that they follow circles on the spherical image [2].

Instead, we propose a new technique for automatically estimating the epipolar geometry and rectifying a spherical stereo image pair using the individual pixel displacements (quantified by dense optical flow) between them without any need for

* Dept. of Precision Engineering, The University of Tokyo, Japan
E-mail: pathak@robot.t.u-tokyo.ac.jp
(Received March 31, 2016)
(Revised January 11, 2017)

calibration patterns or assumptions about the direction of displacement. The only necessity is that the two images be at approximately the same orientation by hand and are not at a prohibitively large distance (several meters or so), in order to ensure valid estimation of optical flow vectors. Or, they can first be approximately rectified using a feature-point based approach [6]. Given two spherical images with a small rotation between them, our algorithm can automatically estimate the complete five degree of freedom motion parameters between them and perfectly align them for stereo disparity estimation followed by 3D reconstruction. To this end, we attempt the use of a dense optical flow approach and directly align all the pixels displacements along the epipolar curves. In a non-linear least squares minimization, we rotate one image while continuously re-estimating the pixel displacements (optical flow vectors) till we arrive at an aligned, translational state. At the end of the minimization, we manage to split the optical flow field into its rotational and translational components and output the relative pose.

2. Related Work

There have been many approaches involving spherical stereo vision [3]–[5],[7]–[9]. While [4],[5],[8], and [9] make use of the usual pattern-based rectification, [3] and [7] work under mechanical alignment of the two cameras. They choose to displace their cameras in the vertical direction, which they claim to be accurate. However, this introduces constraints in the kinds of alignment possible. Since the epipolar geometry is of prime relevance to the accuracy of the estimated points (accuracy is the least as we get closer to the epipolar direction), such constraints could be difficult to work with. Point-based methods like [10] and [11] can also be used, but with the small displacements in stereo images they can be quite inaccurate or unstable. Moreover, feature-point matching always poses problems in highly distorted spherical images. Instead, methods like those in [12],[13], and [14] which optimize an error function over every pixel of the image can be much more accurate, especially in spherical images with a large field of view. However, their work only deals with rotation estimation (not the epipolar geometry), and is not completely invariant to camera translations. On similar lines, since we wish to deal with pixel-to-pixel displacements, optical flow becomes the logical choice to quantify them.

Optical flow on spherical cameras has been researched in great detail before. In [15], the authors discuss the kind of optical flow patterns formed on a spherical camera undergoing pure rotation and translation. Spherical cameras contain information from all directions. Based on this, they explain that since any spherical camera motion is a combination of rotation and translation, any optical flow field on the sphere can be separated into the two components. Thus, they pose the epipolar estimation as a pattern recognition problem and theoretically suggest multiple searches for the 5-DoF parameters along three separate axes. This coupling of translation and rotation parameters along three different aces could be quite cumbersome in practice and lead to errors. Several others like [16] and [17] attempted similar approaches to estimate the 5-DoF motion parameters for mobile robots. Our previous research [18] proposed estimation of rotation using optical flow vectors. Meanwhile, [19] proposed mapping optical flow vectors from the unwarped images to the unit sphere and evaluated three different conventional ego-motion estimation techniques to estimate the rotation and translation among three frames.

3. Overview

Making use of the same theoretical basis as explained in [15], we propose a novel single non-linear least squares minimization to estimate the full 5 DoF epipolar geometry. Since spherical images have information from all angles, they can be rotated to any orientation without loss of information. Thus, the basis behind our approach is to continuously rotate the image in a small space around the original orientation while re-estimating the optical flow vectors until we reach an angle at which all the pixels in the two images are aligned along the epipolar lines, as shown earlier in Fig. 2.

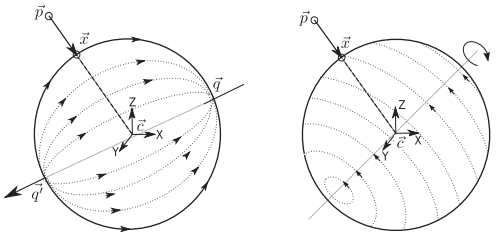
The main contributions of this paper lie in taking advantage of spherical image geometry coupled with dense alignment of individual pixel displacements found by optical flow to estimate the epipolar geometry between two images and rectify them for dense 3D reconstruction. Our approach can precisely estimate the epipolar geometry of a pair of images clicked in a static environment at a translation distance of less than one-fifth, and more than one-twentieth of the average scene distance. Beyond this distance, it was found that the pixel movements become too large to calculate the dense optical flow field correctly. For the same reason, another assumption made in this research is that the images behave similar to a stereo image pair - that they are clicked in the same approximate orientation. This assumption is not too strict as the approximate orientations of the two images can always be found and corrected by sparse feature matching.

In this research, we use the Ricoh Theta camera which automatically provides pre-stitched, completely spherical images using two-oppositely pointed fisheye cameras. The Ricoh Theta has internal calibration of all the required parameters of the two cameras. Thus, we assume that the image is projected on a perfect sphere and there is no need for any further calibration. In the remainder of this paper, we first explain the theoretical basis behind our method, i.e., the patterns of spherical optical flow, the minimization problem, followed by experimental evaluations. We also show and evaluate some dense 3D models generated using our method.

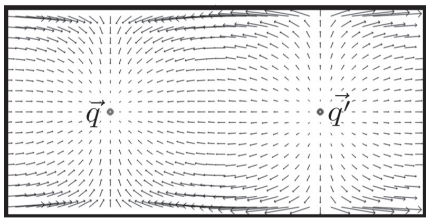
4. Epipolar Geometry and Spherical Motion Fields

Since our spherical camera is displaced in space, it can essentially be reduced to a spherical manifold that has undergone some unknown rotation and translation. This produces certain patterns of optical flow on its surface. As mentioned earlier, spherical motion fields were discussed in great detail in [15] and multi-view geometry for spherical cameras was succinctly explained in [2].

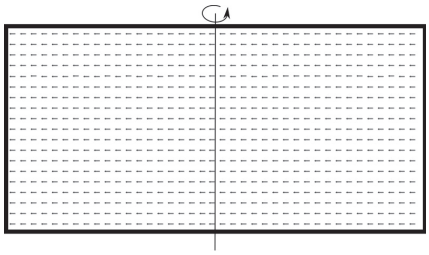
To describe how the flow patterns are formed, we start with the spherical camera model. On the surface of the unit sphere with center $\vec{c}$, a real world point $\vec{P}$ is projected as the intersection of the vector from $\vec{c}$ to $\vec{p}$ with S, as can be seen in Fig. 3. For pure translational motion, the image points projected on the surface of the sphere move on its surface in a tangential direction, diverging away from the epipole $\vec{q'}$ and converging towards the diametrically opposite epipole $\vec{q}$, in effect, forming a 'source' and 'sink' of the optical flow. Thus, they move along *epipolar circles* joining $\vec{q'}$ and $\vec{q}$. As for pure rotational motion,



(a) Translation (spherical) (b) Rotation (spherical)

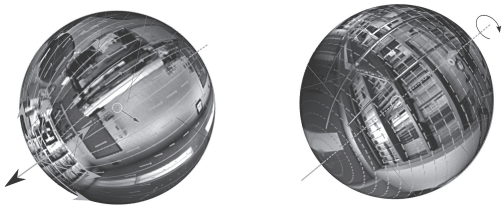


(c) Translation (equirectangular projection). The epipoles $\vec{q'}$ and $\vec{q}$ can be seen.

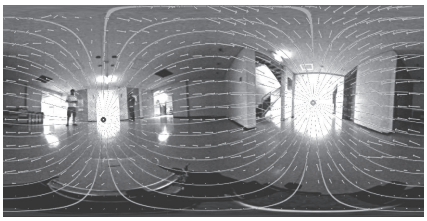


(d) Rotation (equirectangular projection) The axis of rotation is in the vertical direction, while the arrows indicates direction.

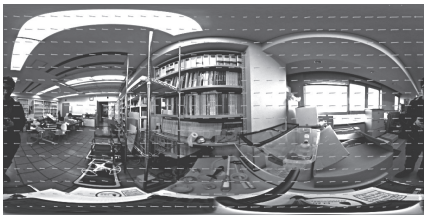
Fig. 3 Motion fields on the unit sphere for the camera undergoing (a) pure translation (b) pure rotation. The arrows indicate the direction of translation and rotation, respectively.



(a) Translation (spherical) (b) Rotation (spherical)



(c) Translation (equirectangular projection). As in Fig. 3, the epipoles $\vec{q'}$ and $\vec{q}$ can also be seen. The curves indicate the epipolar curves. It can be seen that all the optical flow vectors are aligned along the epipolar curves.



(d) Rotation (equirectangular projection) The axis of rotation is in the vertical direction, while the arrow indicates direction, same as Fig. 3

Fig. 4 Real motion fields on the spherical image for (a)(c) translation (b)(d) rotation in both spherical and equirectangular projection. Patterns similar to Fig. 3 above can be noticed, confirming the expected behavior of optical flow.

the optical flow vectors form along loops perpendicular to the axis of rotation. Figure 3 shows both these patterns in both the spherical and equirectangular projections.

For a perfectly aligned pair of spherical images with no rotation between them, the optical flow vectors will be aligned along the epipolar curves (Fig. 3, Fig. 2) and the search for corresponding pixels to calculate disparity can be restricted to these epipolar circles. In order to confirm these patterns, we computed optical flow patterns using artificially generated movement. In one case, the spherical camera was moved sideways (to induce only translation), and in the second, it was rotated on its axis (to induce only rotation). Figure 4 shows the actual optical flow vectors computed on these movements. They were calculated on the equirectangular image using the recently developed DeepFlow algorithm [20] and projected on the surface of the sphere by using the Jacobian of the transformation between the equirectangular image and the unit sphere. The similarity to the patterns in Fig. 3 is clearly visible. The alignment of the optical flow vectors along the epipolar curves (Fig. 4 (c)) can be seen. The epipoles $\vec{q'}$ and $\vec{q}$, corresponding to Fig. 3 can also be noticed.

Any arbitrary displacement of the cameras forms a combination of rotation and translation. In addition to the unknown direction $\vec{q'} - \vec{q}$, there is also a rotation between them, causing a misalignment of optical flow vectors. With a spherical camera, since we can theoretically obtain the flow from any direction, we should be able to uniquely distinguish between a translational and rotational field. Moreover, as mentioned before, rotations in spherical images are completely recoverable. Hence, it should theoretically be possible to 'derotate' the image to an orientation at which all the optical flow vectors are aligned along an epipolar direction. Our approach relies on this principle posed as a non-linear least squares minimization to obtain the precise epipolar geometry between two images. Thus, we continuously 'derotate' one image with respect to the other and re-estimate the optical flow vectors between them while minimizing an error, till we reach a state resembling pure translation in a particular direction, which forms the epipolar direction. In such a state, the images can be said to be rectified and the stereo disparity estimation can be done easily by searching along the epipolar curves. In essence, our algorithm tries to directly align the pixel displacements (characterized by optical flow) along epipolar curves by minimizing their angle with the epipolar curves over the 5 DoF epipolar parameters.

[21] also used a similar concept of aligning pixel displacements. However, their approach minimizes an error in feature point matching. Moreover, they only performed numerical experiments. In the same manner as [12],[13], and [14], it would be more stable to involve all the pixels of the image in calculating the error. In the next section, we explain our motion model and the formulation of the error function.

5. Epipolar Alignment by Minimization

5.1 Notation and Motion Model

First, we describe our motion model with respect to spherical images. For rotation, we define three parameters α, β, and γ along the three $x-y-z$ Euler axes. Thus, the rotation matrix between two images becomes $R = R_x(\alpha)R_y(\beta)R_z(\gamma)$, where R_x, R_y, and R_z are the individual rotations along the three axes X, Y, and Z respectively. The reason for this choice of representing rotations is that the rotations between stereo images are typically small and Euler angles with fixed axes ensure a small search space as opposed to a quaternion based approach. For every point on the sphere S, we follow the cartesian coordinate notation and denote the radius vector of the point as $\vec{x}$, with the origin being $\vec{c}$, the center of the sphere. Meanwhile, we indicate the translation direction by the one of the epipoles $\vec{q}$ (the point opposite to direction of motion of the sphere). Only as an input to the optimization, we denote $\vec{q}$ with spherical coordinates θ and ϕ to ensure that all five parameters take the form of angles. Doing so ensures uniform scaling of all parameters in the upcoming non-linear least squares minimization. Thus, the 5 DoF parameter vector $G(\alpha, \beta, \gamma, \theta, \phi)$ uniquely defines the epipolar geometry between two spherical images.

Meanwhile, the optical flow vector at $\vec{x}$ is written as $\vec{F}$ (relative to $\vec{x}$). $\vec{F_{trans}}$ is defined as the component of optical flow caused by translation, which is tangential to the great circle connecting $\vec{x}$ to the epipole $\vec{q}$ (i.e. the epipolar curve). $\vec{F_{rot}}$ is the component caused by rotation, tangential to the circle passing through $\vec{x}$, perpendicular to the axis of rotation. All notations can be seen in Fig. 5.

5.2 Error Formulation

To summarize the previous subsection, our optimization space consists of 5 parameters: α, β, and γ for rotation, and θ and ϕ for translation. Next, we come to the error formulation. From the previous section, we can conclude that there will always be an orientation to which an image can be rotated at which all the pixels are displaced along an epipolar direction, resulting in a pure translational displacement. We find this orientation by minimizing the angle between the epipolar lines and pixel displacements (optical flow).

In each iteration, we first use the current estimates of the rotation angles α, β, and γ along the three $x-y-z$ Euler axes to 'derotate' the image to a purely translational displacement with respect to the other image. In this state, we estimate the dense optical flow field, i.e., calculate all optical flow vectors $\vec{F}$. In our research, we made use of the Deepflow algorithm [20] to estimate the dense optical flow field. It would be very time consuming to do so in every iteration. Hence, we use a simple reprojection based on the initial dense optical flow field, as described later in Section 5.3.

In case of correct estimation of all parameters, the optical flow field should be purely translational, and for each point $\vec{x}$ on the sphere the optical flow vector $\vec{F}$ should be tangential to the epipolar curve, i.e., the great circle connecting it to the epipole $\vec{q}\,(\theta, \phi)$. In other words, $\vec{F_{rot}} = 0$ and $\vec{F} = \vec{F_{trans}}$ ideally. Therefore, we can define the angular difference between $\vec{F}$ and $\vec{F_{trans}}$ as the error at point $\vec{x}$, denoted by Ω. Ω can be calculated as follows (as shown in Fig. 6):

Since our optical flow vector $\vec{F}$ and the 'expected' optical flow vector $\vec{F_{trans}}$ are both tangential to great circles (epipolar curves) on the spherical image, we need to find the angle between these circles. We first find the great circle C_q to which $\vec{F_{trans}}$ is tangential, defined by $\vec{x}$ and the epipole $\vec{q}$. For each great circle on the sphere S, there exists a unique normal vector emanating from the centre of the sphere. The normal vector $\vec{N_q}$ for the epipolar great circle C_q defined by $\vec{x}$ and epipole $\vec{q}$ can be found by taking the cross product between them:

$$\vec{N_q} = \vec{q} \times \vec{x}. \tag{1}$$

Thus, $\vec{N_q}$ is the normal vector to the great circle C_q along $\vec{F_{trans}}$ that the optical flow vector is 'expected' to follow. The actual optical flow vector $\vec{F}$ also follows a great circle C_f on the sphere, which can be said to be defined by $\vec{x}$ and $(\vec{x}+\vec{F})$. Thus, to find its normal vector N_f, we again take their cross product:

$$\vec{N_f} = (\vec{x} + \vec{F}) \times \vec{x}. \tag{2}$$

Now, the angle Ω between the expected normal vector $\vec{N_q}$ and the actual normal vector $\vec{N_f}$ becomes:

$$\Omega = \arccos\left(\frac{\vec{N_f} \cdot \vec{N_q}}{|\vec{N_f}||\vec{N_q}|}\right). \tag{3}$$

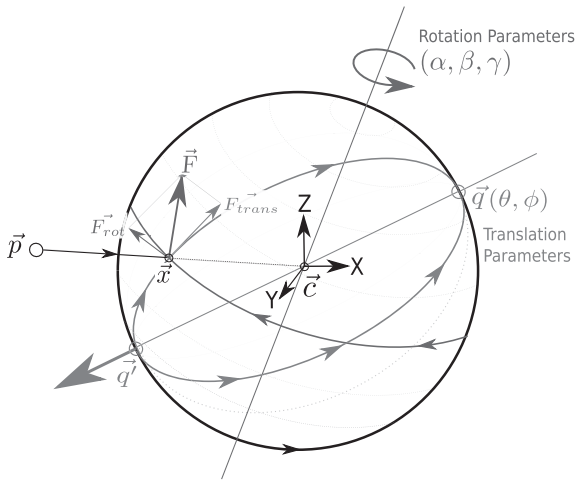


Fig. 5 Motion model and notations. All translation parameters are represented by epipole $\vec{q}(\theta, \phi)$. All rotation parameters are represented by $< x-y-z >$ euler angles α, β, and γ. $\vec{F_{trans}}$ is the component of optical flow caused by translation while $\vec{F_{rot}}$ is the component due to rotation. The total optical flow vector $\vec{F}$ is $\vec{F_{trans}} + \vec{F_{rot}}$.

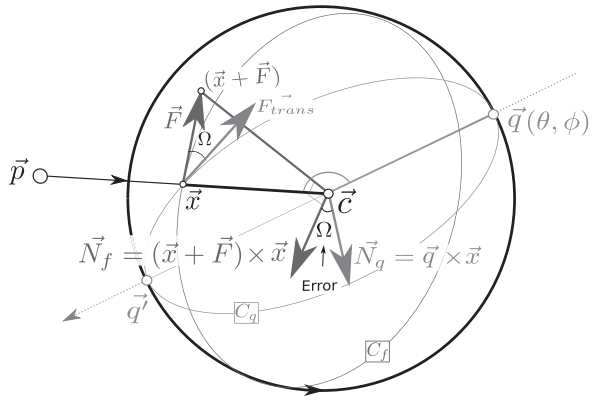


Fig. 6 Calculating the error at $\vec{x}$. To find angle Ω, we take the cross products $(\vec{x}+\vec{F}) \times \vec{x}$ and $\vec{q} \times \vec{x}$ in order to define the great circles along $\vec{F_{trans}}$ and $\vec{F}$ and find the angle between them.

To summarize the error formulation in each iteration for the 5 DoF parameter vector $G(\alpha,\beta,\gamma,\theta,\phi)$, the steps are:

- 'De-rotate' the image with euler angles α, β, and γ
- Estimate optical flow vector $\vec{F}$ at every point $\vec{x}$
- According to $\vec{q}(\theta,\phi)$, estimate the angle between expected displacement $\vec{F_{trans}}$ and actual displacement $\vec{F}$ for every pixel $\vec{x}$
- Error at point $\vec{x}$ is taken to be this angle Ω (calculated from $\vec{N_f}$ and $\vec{N_q}$).

This error is used in a non-linear least squares minimization. Combining (1), (2), and (3), the final optimization is posed as follows in (4):

$$\underset{G(\alpha,\beta,\gamma,\theta,\phi)}{\text{minimize}} \sum_{\forall \vec{x_i}} \left(\arccos\left(\frac{(\vec{q}\times\vec{x})\cdot((\vec{x}+\vec{F})\times\vec{x})}{|(\vec{q}\times\vec{x})||((\vec{x}+\vec{F})\times\vec{x})|} \right) \right)^2, \quad (4)$$

where $\vec{F}$ has been calculated after de-rotation with (α,β,γ), which can be done via a simple reprojection (next subsection), and $\vec{q}$ is the epipolar point (θ,ϕ) on the sphere (converted to cartesian coordinates). For the purpose of this research, the popular Levenberg-Marquardt algorithm [22] was adopted.

5.3 Reprojection of Optical Flow

In our approach, the error function and its Jacobian with respect to the 5 DoF parameter vector $G(\alpha,\beta,\gamma,\theta,\phi)$ can only be calculated after rotating one image with respect to the other and re-estimating the optical flow. This makes every iteration of the optimization quite tedious.

To solve this issue, we propose a simplistic reprojection of the optical flow vectors based on the initial state between the two frames. Given the initial optical flow vector $\vec{F}$ at $\vec{x}$ on the sphere, our objective is to calculate the optical flow vector $\vec{F_r}$ at any iteration, given the rotation angles α,β, and γ. Given that $\vec{x}$ lies on one image (the one that is rotated) and $\vec{x}+\vec{F}$ is its corresponding point in the other image, we rotate $\vec{x}$ with the rotation matrix R composed as $R_x(\alpha)R_y(\beta)R_z(\gamma)$ as shown in (5):

$$\vec{x_r} = R_x(\alpha)R_y(\beta)R_z(\gamma)\times\vec{x}. \quad (5)$$

Since the point $\vec{x}+\vec{F}$ on the second image is unchanged, the new optical flow vector $\vec{F_r}$ can be calculated as:

$$\vec{F_r} = (\vec{x}+\vec{F}) - \vec{x_r} = (\vec{x}+\vec{F}) - R_x(\alpha)R_y(\beta)R_z(\gamma)\times\vec{x}. \quad (6)$$

A visualization of this reprojection is shown in Fig. 7.

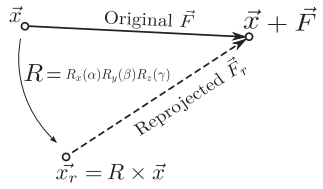
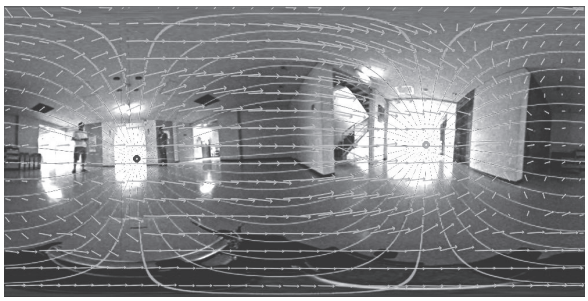


Fig. 7 Reprojection of the optical flow vector given a small rotation α,β,γ.

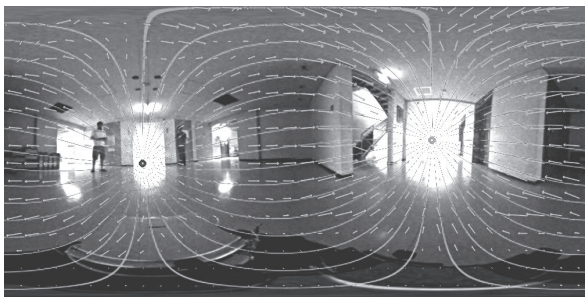
5.4 Initial Estimate and Choice of Optimization Space

For most non-linear least squares minimization, a good initial value of the parameter vector, close to the optimum, is required. For our approach, the initial solution can be obtained using sparse corresponding points, i.e. the common 8-point algorithm [6]. However, the initial solution can be provided by the initial dense optical flow field itself. Since the dense optical flow field is also used in estimating the error in the minimization, this approach reduces computation time and provides a simpler solution to estimate all required information.

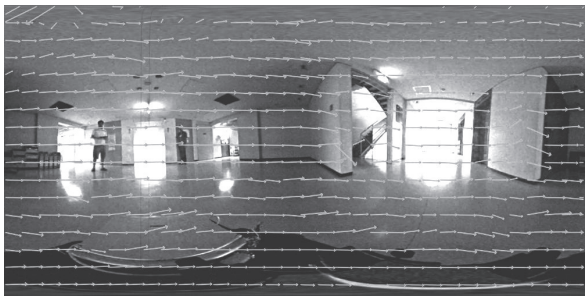
Since the two images in a stereo pair are taken from similar orientations (or after correction using sparse feature information), the initial value for rotation (α,β,γ) can be simply set to zero. The other reason for this necessity is that optical flow vectors cannot be computed properly if the difference between the two images is too large. Regarding the initial value of the epipole (ϕ,θ), we chose to take the vector sum of all optical flow vectors in the initial state. The optical flow pattern is a vector sum of rotation and translation, and the vector sum of the rotational vectors is zero because of symmetry (as in Fig. 3). Thus, the vector sum of the entire field will ignore rotation and approximately lie near the epipole (θ,ϕ), which suffices as a good



(a) Before optimization: Translational plus rotational flow. Initial estimates of epipoles $\vec{q}$ and $\vec{q'}$ and epipolar curves.



(b) After optimization: Translational flow vectors aligned along epipolar curves. Final estimates of epipoles $\vec{q}$ and $\vec{q'}$.



(c) After optimization: Subtracting the translational flow from (a) gives us the rotational flow.

Fig. 8 Optical flow field on the equirectangular projection of the spherical images (a) before optimization (b) after optimization: translational flow (c) after optimization: rotational flow (similar to Fig. 4).

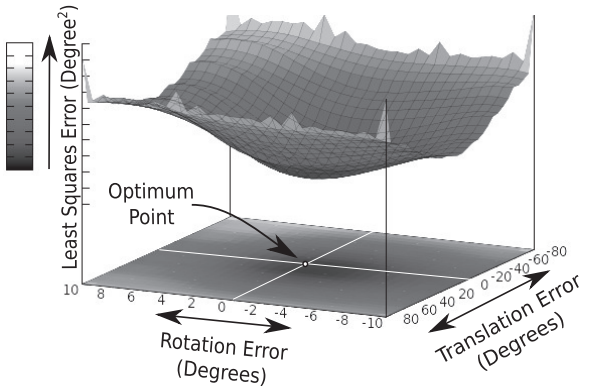


Fig. 9 Error function with respect to translation direction (epipole) and rotation errors, i.e. deviations from the optimum.

initial value for the translation direction.

5.5 Output

The optimization outputs the best value of the parameter vector for which the optical flow vectors appears to be of only translation between the two images. As mentioned earlier, we essentially try to simultaneously optimize the rotation and the epipolar direction by directly aligning pixel displacements (optical flow) along the epipolar curves. Figure 8 shows an example of this optimization. Two images as a stereo pair were clicked and the proposed approach was used to process them. The initial optical flow field is rotation plus translation and the optical flow vectors are unaligned. After optimization, the entire field is aligned along the epipolar curves, as in pure translation. Finally, subtracting the optimized optical field from the initial field results in a rotational optical flow field, confirming that minimization worked.

Meanwhile, the translation direction and rotation were gradually varied in steps from the estimate obtained from the solution of the proposed minimization between these images. The translation direction was varied in a randomly chosen direction and the rotation angle was varied about a randomly chosen axis for the sake of simplicity and ease of visualization. The error value (see (4)) was evaluated at every step and is displayed as a graph in Fig. 9. The translation error and rotation error are, respectively, the deviations from the final optimum estimate of translation direction and rotation. It can be seen that the convex shape of this function with a single minimum present at the location where both errors are zero is suitable for a minimization. Once we obtain this epipolar geometry, we calculate the disparity by searching for the displacement along the epipolar curves using the previously mentioned DeepFlow algorithm [20]. Following this, the 3D position of each pixel can be triangulated in a manner similar to that given in [3].

6. Experimental Evaluation

Two experiments were performed to check the accuracy of the estimated epipolar geometry. Experiment 1 was done to check the accuracy of the estimated epipolar direction vectors, while Experiment 2 was performed to check the quality of the 3D reconstruction using the estimated epipolar geometry.

6.1 Experiment 1

Experiment 1 was done to check the accuracy of the estimated epipolar direction vectors using a known configuration

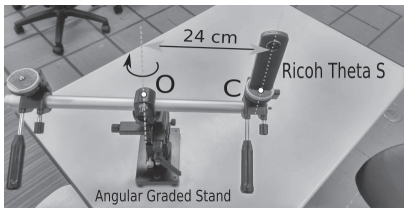
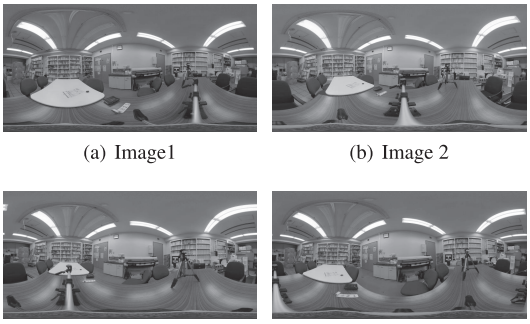


Fig. 10 Experiment 1: experimental setup.



(a) Image1 (b) Image 2

(c) Image 3 (d) Image 4

Fig. 11 Experiment 1: 4 stereo images captured as shown in Fig. 12.

of camera positions as groundtruth. Four images were captured at approximately the same orientation (after rough adjustments by hand) in a known geometric configuration using a graded camera stand (with marked angles and distances) and the epipolar direction vector between each pair of images was estimated using the optimization. The camera (Ricoh Theta S) was placed at the end of a 24 cm long arm with the other end attached to a graded camera stand (with marked angles), as seen in Fig. 10. The arm was rotated counter-clockwise around point O. At every 90°, the camera was adjusted to the same approximate orientation and a single image was captured. Four images (shown in Fig. 11) were captured at locations $C_{1,2,3,\text{ and }4}$, resulting in a square configuration, as shown in Fig. 12 (a). The epipolar direction vectors ($e_{i,j}$, where $i, j \in 1, 4$) between all pairs of images were estimated using the optical-flow based optimization described.

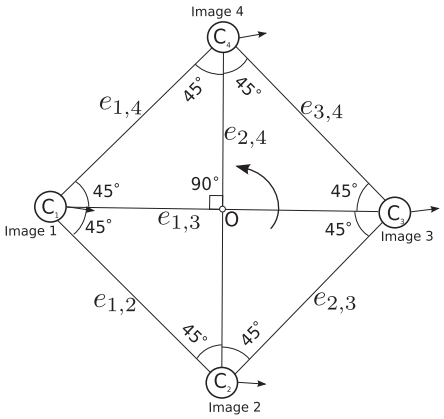
At each image, the angles between the epipolar direction vectors with respect to the other 3 images were calculated and compared to the groundtruth (Fig. 12). For example, at Image 3, the angles between epipolar direction vectors $e_{2,3}$ and $e_{1,3}$, and vectors $e_{1,3}$ and $e_{3,4}$ were estimated (estimating the angles between $e_{3,4}$ and $e_{2,3}$ becomes redundant).

Each angle should be 45°, as visible from the configuration. The results are shown in Table 1 and Fig. 12 (b). The mean error was **0.51°** indicating that the estimation was accurate.

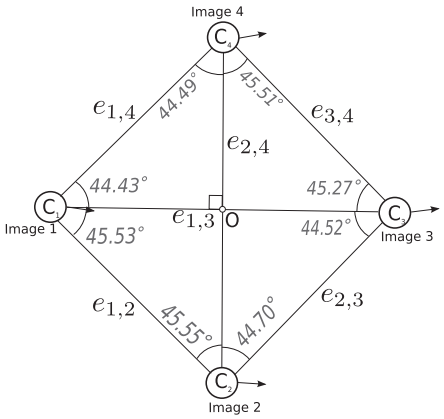
6.2 Experiment 2

Experiment 2 was conducted in an artificially constructed environment to check for the quality of 3D reconstruction. Two square slabs of cardboard (1 m×1 m) were placed at 90° to each other and the camera set on a graded camera stand, as shown in Fig. 13. One image was captured at the original position. Next, the camera was translated towards its left by 5.5 cm and rotated counter-clockwise around its axis by precisely 6°. In this position, the second image was captured (shown in Fig. 14).

The algorithm was applied to this pair of images. Stereo disparity along the epipolar lines was estimated using DeepFlow [20] and the 3D structure was reconstructed by triangula-



(a) Four images captured in known configuration, after rotating the arm 90°. around O. Angles between all epipolar direction vectors are shown (groundtruth)

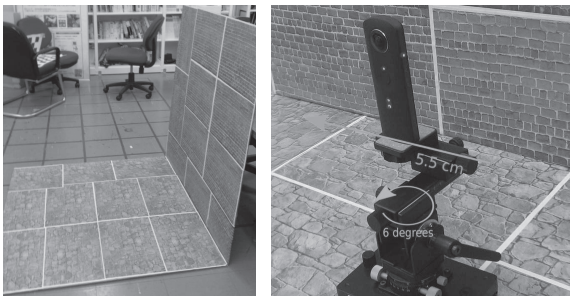


(b) Estimated angles between epipolar direction vectors of all images

Fig. 12 Experiment 1: experimental configuration and results of epipolar estimation.

Table 1 Results of estimation of epipolar direction vectors.

Image	Angle	Groundtruth	Estimated Result
Image 1	$e_{1,2} - e_{1,3}$	45°	45.53°
	$e_{1,3} - e_{1,4}$	45°	44.43°
Image 2	$e_{1,2} - e_{2,4}$	45°	45.55°
	$e_{2,4} - e_{2,3}$	45°	44.70°
Image 3	$e_{3,4} - e_{1,3}$	45°	45.27°
	$e_{1,3} - e_{2,3}$	45°	44.52°
Image 4	$e_{1,4} - e_{2,4}$	45°	44.49°
	$e_{2,4} - e_{3,4}$	45°	45.51°



(a) Measurement setup (b) Graded camera stand with spherical camera

Fig. 13 Experiment 2: experimental setup for 3D reconstruction.

tion [3]. Figure 15 shows the results of the algorithm on these two images, namely, the original optical flow, the separated rotational and translational flows, and the disparity map obtained.

(a) Image at original position

(b) Image after translating left by 5.5 cm and rotation by 6°

Fig. 14 Experiment 2: spherical stereo image pair of the setup shown in Fig. 13.

Table 2 Rotation angle error and average planar deviation in Experiment 2.

Error in Rotation Angle	Average Planar Deviation
0.05°	0.51%

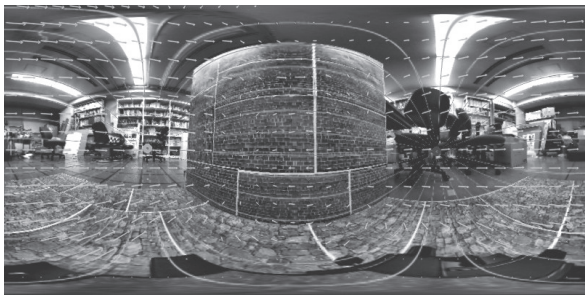
Table 3 Calculation times for various steps in Experiment 2.

Process	**Average time**
Initial optical flow estimation	36.20 s
Optimized optical flow estimation	18.5 s
Disparity Estimation using DeepFlow [20]	36.25 s
3D structure [3]	0.28 s
Total	**91.13 s**

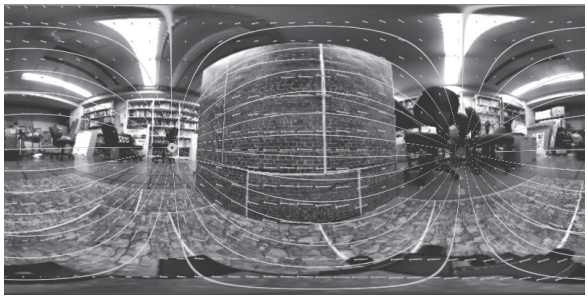
The rotation angle was found to be **6.05°**, which indicates an error of 0.05°. The results of the reconstruction are shown in Fig. 16. One perspective view and the side view is shown next to the actual structure for comparison and the similarity with the original structure can be seen, thus proving that this algorithm is suitable for dense 3D reconstruction. In order to check the quality of the 3D reconstruction, the planarity of the two planes in the structure was calculated separately by plane fitting and the average planar deviation of the structure (from both planes) was found as a percentage of the length of the side. It was found to be 0.51% of the side length. Both the rotation angle estimation error and planar deviation are reported below in Table 2.

6.3 Calculation Time

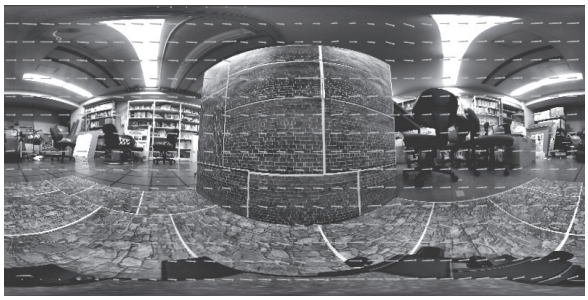
All the processing was done on a computer equipped with a 2.8 GHz Intel Xeon processor, without any parallel processing. The calculation times obtained for various steps in Experiment 2 are given below in Table 3. The image resolution used was 1500×750 pixels. The most time-consuming step of the process is the deepflow algorithm used for dense optical flow computation, which occurs twice in the process. It takes very long for a higher image resolution. A possible workaround could be to estimate the epipolar geometry (initialization and optimization)



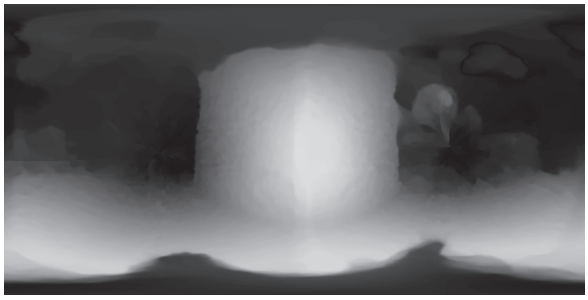
(a) Before optimization: Initial optical flow pattern and epipolar state with unaligned optical flow vectors.



(b) After optimization: Final optical flow pattern and epipolar state with optical flow vectors aligned along epipolar curves: Translational flow



(c) After optimization: Translational flow vectors subtracted from the initial state, reprojected back on the image. Rotational flow



(d) Disparity estimation along epipolar curves using the Deepflow [20] algorithm

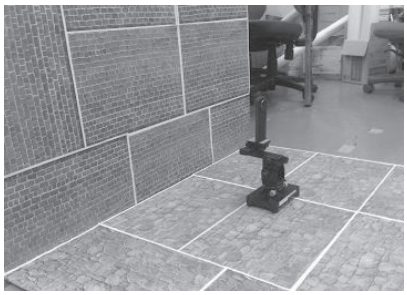
Fig. 15 Experiment 2: output of the algorithm applied to the image pair in Fig. 14.

at a lower resolution, and the disparity estimation and 3D reconstruction at a higher. A further speed-up can be obtained by using a GPU for this step.

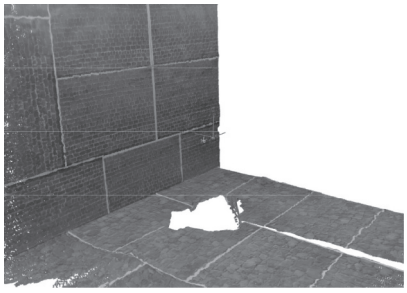
7. Discussion

7.1 Factors Affecting Accuracy and Limitations of Estimation

The main factors that affect the accuracy of the estimation are as follows - 1) The presence of textures in the environment - This is because textures help in the estimation of accurate dense optical flow, on which our method is based. Without sufficient



(a) Experiment setup



(b) Reconstructed view showing the camera position

(c) Reconstructed side view

Fig. 16 Experiment 2: 3D reconstructed views of the setup from Fig. 13.

textures, the estimation of dense optical flow becomes ambiguous and unstable. 2) The type of camera translation - This forms an important limiting factor for our approach. Dense optical flow computation is based on the assumption that pixel movements are small. Thus, if the translation of the camera is too large, the pixel movement becomes too large and it can induce errors in the measurement. It was observed empirically that if the distance moved was more than around one-fifth of the average scene distance, the pixel movements became too high to compute dense optical flow properly. Thus, neither epipolar estimation nor reconstruction are possible. On the other hand, if the distance moved was less than one-twentieth, it was observed that the optical flow vectors were too small and affected by noise to compute any meaningful reconstruction. However, epipolar estimation was still possible in this case. Conversely, due to the same reason, accurate 3D reconstruction was possible for all regions located at a distance of within 5 times to 20 times the translation distance of the camera.

Another factor that can affect the accuracy is a camera performance limitation. In this research, we assume that all points are projected to a perfect sphere. However, a small distance exists between the two lenses of the Ricoh Theta camera causing points too close to the camera - within 20 cm, as per our observations - to be distorted. Thus, points too close to the camera

can induce errors and cannot be reconstructed properly.

It is also important to note that 3D reconstruction errors become large near the camera stand, as can be seen in Fig. 16 (b). This is because of multiple reasons. First, the portion occluded by the camera stand cannot be seen by the spherical camera. Second, our approach uses the movement of pixels as computed by dense optical flow. For points too close to the camera, pixel movement is too large to be accurately estimated by dense optical flow which tries to constrain the movement of pixels. Moreover, as explained earlier, a slight gap exists between the centers of the two individual fisheye cameras. This does not affect points sufficiently away from the camera. However, points very close to the camera center (within 20 cm) will not be projected to a perfect sphere and can be distorted.

7.2 Applications

One of the applications being considered for this research is the inspection of large infrastructures. Large infrastructures like bridges, dams, etc. need to be observed from a very close distance to check for cracks or other defects. Due to their large size, many approaches have been proposed using flying robots or drones equipped with cameras. These drones can fly close to the structure and inspect it with much more ease as compared to a human. A 3D model of the infrastructure can be constructed and inspected offline on a computer.

In such cases, a perspective camera that can only see a small portion of the large structure will have difficulty in tracking image information. A spherical camera that can observe the entire structure at once can be very effective. A video can be continuously recorded by a spherical camera and keyframes satisfying the limitations of the proposed approach can be selected by measuring the average pixel movements. An accurate dense 3D reconstruction can be constructed and merged together in a visual-SLAM like approach.

With regards to localization, we showed an average localization error of 0.51° with respect to two image localization in Experiment 1. That corresponds to an error or around 0.89%, obtained by converting the angle to radians and multiplying by 100). Meanwhile, in Experiment 2, the rotation error was 0.05°. With regards to 3D reconstruction, the error obtained was a planar deviation of 0.51%. These are suitable for practical application. Further experiments are necessary to evaluate the accuracy in a continuously moving video. The calculation time of around 91 s is high, but there is no requirement for online computation for this application. Further speed-ups can be obtained by using a GPU for optical flow computation.

8. Conclusion

In this paper, we have shown a method to automatically estimate the epipolar geometry for spherical image pairs. The main contribution of our paper is to perform this estimation using an error minimized directly over dense pixel displacements (optical flow) in the entire image, instead of just a few keypoints. Taking advantage of spherical image geometry, we showed how a spherical flow field can be split into its rotational and translational components. As indicated by Experiment 1, the estimation of the epipolar geometry was quite accurate (std. dev. of error = 0.51°). 3D reconstruction was also demonstrated in Experiment 2.

The method is especially suited for dense 3D reconstruction as it directly aligns all the pixels in the image along the epipolar curves. We can generate accurate dense disparity maps without fixing two cameras or calibration. We demonstrated this in Experiment 2 by showing the densely reconstructed 3D model. In future, we would like to extend this approach to process a full video, frame-by-frame.

Acknowledgments

This work was in part supported by Council for Science, Technology and Innovation, "Cross-ministerial Strategic Innovation Promotion Program (SIP), Infrastructure Maintenance, Renovation, and Management" (funding agency: NEDO).

References

[1] R.I. Hartley and A. Zisserman: *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[2] A. Torii, A. Imiya, and N. Ohnishi: Two-and three-view geometry for spherical cameras, *Proceedings of the Sixth Workshop on Omnidirectional Vision, Camera Networks and Non-Classical Cameras*, pp. 81–88, October 2005.

[3] H. Kim and A. Hilton: 3d scene reconstruction from multiple spherical stereo pairs, *International Journal of Computer Vision*, Vol. 104, No. 1, pp. 94–116, August 2013.

[4] S. Li: Real-time spherical stereo, *Proceedings of the International Conference on Pattern Recognition*, Vol. 3, pp. 1046–1049, August 2006.

[5] S. Li: Binocular spherical stereo, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 9, No. 4, pp. 589–600, December 2008.

[6] R.I. Hartley: In defense of the eight-point algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 6, pp. 580–593, June 1997.

[7] H. Kim and A. Hilton: Planar urban scene reconstruction from spherical images using facade alignment, *Proceedings of the IEEE IVMSP Workshop*, pp. 1–4, June 2013.

[8] N. Kita: Dense 3d measurement of the near surroundings by fisheye stereo, *Proceedings of the IAPR Conference on Machine Vision Applications*, pp. 148–151, June 2011.

[9] N. Kita, F. Kanehiro, and Y. Kita: 3d shape measurement of a large cloth close to a fisheye stereo, *Proceedings of the IEEE/SICE International Symposium on System Integration (SII)*, pp. 895–900, Dec. 2012.

[10] Z. Kukelova, J. Heller, M. Bujnak, A. Fitzgibbon, and T. Pajdla: Efficient solution to the epipolar geometry for radially distorted cameras, *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2309–2317, December 2015.

[11] D.V. Papadimitriou and T.J. Dennis: Epipolar line estimation and rectification for stereo image pairs, *IEEE Transactions on Image Processing*, Vol. 5, No. 4, pp. 672–676, April 1996.

[12] A. Makadia, L. Sorgi, and K. Daniilidis: Rotation estimation from spherical images, *Proceedings of the International Conference on Pattern Recognition*, pp. 590–593, August 2004.

[13] A. Makadia and K. Daniilidis: Rotation recovery from spherical images without correspondences, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 7, pp. 1170–1175, August 2006.

[14] A. Makadia and K. Daniilidis: Direct 3d-rotation estimation from spherical images via a generalized shift theorem, *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 217–224, June 2003.

[15] R.C. Nelson and J. Aloimonos: Finding motion parameters from spherical motion fields (or the advantages of having eyes in the back of your head), *Biological Cybernetics*, Vol. 58, No. 4, pp. 261–273, March 1988.

[16] T.W. Hui and R. Chung: Determining motion directly from normal flows upon the use of a spherical eye platform, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2013*, pp. 2267–2274, 2013.
[17] A. Briod, J.C. Zufferey, and D. Floreano: A method for ego-motion estimation in micro-hovering platforms flying in very cluttered environments, *Autonomous Robots*, Vol. 40, Issue 5, pp. 789–803, 2016.
[18] S. Pathak, A. Moro, A. Yamashita, and H. Asama: Rotation removed stabilization of omnidirectional videos using optical flow, *Proceedings of the 6th International Conference on Advanced Mechatronics*, pp. 51–52, December 2015.
[19] J. Gluckman and S.K. Nayar: Ego-motion and omnidirectional cameras, *Proceedings of the Sixth International Conference on Computer Vision 1998*, pp. 999–1005, 1998.
[20] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid: DeepFlow: Large displacement optical flow with deep matching, *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1385–1392, December 2013.
[21] J. Fujiki, A. Torii, and S. Akaho: Epipolar geometry via rectification of spherical images, *Proceedings of the Third International Conference on Computer Vision/Computer Graphics Collaboration Techniques*, pp. 461–471, March 2007.
[22] M.I.A. Lourakis: levmar: Levenberg-marquardt nonlinear least squares algorithms in c/c++, http://www.ics.forth.gr/~lourakis/levmar/, July 2004.

Sarthak Pathak

He received his Bachelor of Technology and Master of Technology degrees from the Department of Engineering Design, Indian Institute of Technology, Madras (IITM) in 2014. He is currently a second year doctoral student in the department of precision engineering, the University of Tokyo. His research interests include omnidirectional vision, especially 3D reconstruction, SLAM, and stereo vision using them.

Alessandro Moro

He is a Research Engineer at Ritecs Inc. and a Visiting Research Fellow on Computer Vision at the University of Tokyo. He graduated in Computer Science at the University of Udine (Italy) in 2006 and received his Ph.D. from the University of Trieste (Italy) in 2011. His research interests span computer and human vision, computer graphics, and machine learning. His research is in the areas of computer vision, objects recognition, and 3D reconstruction. Main interests are object recognition and machine learning for robotic application.

Atsushi Yamashita (Member)

He received his B.E., M.E., and Ph.D. degrees from the Department of Precision Engineering, the University of Tokyo, Japan, in 1996, 1998, and 2001, respectively. From 1998 to 2001, he was a Junior Research Associate in the RIKEN (Institute of Physical and Chemical Research). From 2001 to 2008, he was an Assistant Professor of Shizuoka University. From 2006 to 2007, he was a Visiting Associate of California Institute of Technology. From 2008 to 2011, he was an Associate Professor of Shizuoka University. From 2011, he is an Associate Professor in the Department of Precision Engineering, the University of Tokyo. His research interests include robot vision, image processing, multiple mobile robot system, and motion planning. He is a member of ACM, IEEE, JSPE, RSJ, IEICE, JSME, IEEJ, IPSJ and ITE.

Hajime Asama (Member)

He received his B.S., M.S., and Dr.Eng. degrees from the University of Tokyo, Japan, in 1982, 1984, and 1989, respectively. He was Research Associate, Research Scientist, and Senior Research Scientist in RIKEN (The Institute of Physical and Chemical Research, Japan) from 1986 to 2002. He became a professor of RACE (Research into Artifacts, Center for Engineering), the University of Tokyo in 2002, and a professor of School of Engineering, the University of Tokyo since 2009. Currently, he is the chairman of the Task Force for Remote Control Technology of the Council for the Decommissioning of TEPCO's Fukushima Daiichi NPS, the leader of Project on Disaster Response Robots and Their Operation System of Council on Competitiveness-Japan, and the chairman of Robotics Task Force for Anti-Disaster (ROBOTAD). His main research interests are distributed autonomous robotic systems, smart spaces, service engineering, and Mobiligence, and service robotics.