Paper:

Spherical Video Stabilization by Estimating Rotation from Dense Optical Flow Fields

Sarthak Pathak, Alessandro Moro, Hiromitsu Fujii, Atsushi Yamashita, and Hajime Asama

The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
E-mail: {pathak, moro, fujii, yamashita, asama}@robot.t.u-tokyo.ac.jp

[Received November 25, 2016; accepted February 2, 2017]

We propose a method for stabilizing spherical videos by estimating and removing the effect of camera rotation using dense optical flow fields. By derotating each frame in the video to the orientation of its previous frame in two dense approaches, we estimate the complete 3 DoF rotation of the camera and remove it to stabilize the spherical video. Following this, any chosen area on the spherical video (equivalent of a normal camera's field of view) is unwarped to result in a 'rotation-less virtual camera' that can be oriented independent of the camera motion. This can help in perception of the environment and camera motion much better. In order to achieve this, we use dense optical flow, which can provide important information about camera motion in a static environment and can have several advantages over sparse feature-point based approaches. The spatial regularization property of dense optical flow provides more stable motion information as compared to tracking sparse points and negates the effect of feature point outliers. We show superior results as compared to using sparse feature points alone.

Keywords: video stabilization, spherical videos, dense optical flow

1. Introduction

Spherical cameras that can view all directions in space in real-time (such as the Ricoh Theta) have become popular these days. They are often used to provide an immersive view of the entire surrounding environment. Due to these advantages, they have also found use in robotics. Remotely operated robots equipped with spherical cameras can be used for visual surveying in situations dangerous for humans. Often, robots such as drones have unstable motion and looking at the spherical videos directly can lead to user disorientation. It is especially difficult for the viewer to concentrate on the survey. In such cases, stabilizing the video by estimating the camera rotation could help immensely. Since most of the instability and disorientation in a video is due to camera rotation, it is enough to nullify it and create a 'rotation-less virtual camera.' Previously, [1] and [2] talked about the development of such a rotation-less camera which can be oriented at will independent of the robot's motion. They experimentally concluded that it could be immensely useful in visual surveying and serve to reduce confusion.

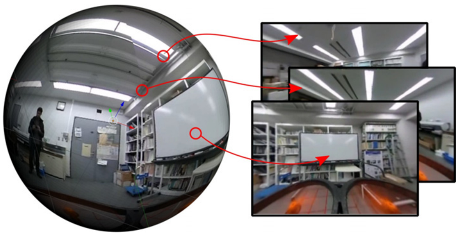


Fig. 1. Spherical images contain information from all directions and can be unwarped at any point to give a perspective view. Thus, they can provide an immersive view of the surroundings.

A major advantage of spherical cameras is that since they contain information from all directions, any area on the image can be unwrapped to give the perspective view in that particular direction, as shown in **Fig. 1**. Moreover, they can also be rotated to any orientation without loss of information. Thus, in this research, we attempt to estimate the rotation of a spherical camera video in order to stabilize it and create such a virtual, rotation-less camera. This is doubly effective as most image-based motion estimation algorithms are more effective using spherical images. We make use of the dense optical flow field information to achieve this.

Tracking the motion of a camera through images/videos has been addressed since long. The basic concept is to track the flow of a number of pixels and back-calculate the camera motion using the camera geometry. Tracking sparse feature points can be done by a plethora of feature descriptors like SIFT [3] that can match points across multiple images. However, this matching of sparse points can result in outliers that often cannot be filtered out easily, even by techniques such as RANSAC [4]. These can make motion estimation unstable and affect the accuracy of the result. Traditionally, most work on camera motion estimation uses sparse feature points. In contrast, dense optical flow algorithms attempt to provide the complete motion information of all pixels in the image and are particularly effective in case of videos with small frame-to-frame motions. Since this is an ill-constrained problem, it is often achieved through various forms of spatial regularization and smoothing. This regularization leads to smooth pixel motion estimates that may differ slightly from the true value, but are free of drastic outliers, unlike sparse feature points. In this work, we focus on the use of dense optical flow for our rotation estimation and stabilization.

We continuously estimate camera motion and derotate each frame to the orientation of its previous frame in a dense minimization of the optical flow to convert it to a pure translational motion, thus creating a rotation-less camera. Two types of minimizations based on frame-to-frame motion properties are introduced. One is a 3 DoF approach using only the rotation, and the other is a frame-to-frame 5 DoF full epipolar minimization, both based on dense optical flow. The estimation is superior as compared to using only feature points. We use the inexpensive Ricoh Theta S camera for this research which outputs spherical images directly in an equirectangular format. The properties of the spherical camera are exploited for a two-fold advantage: the information contained in all directions is used effectively in a dense minimization, and it is also used to provide information for the virtual, rotation-less camera.

Next, in Section 2, we talk about other related work in this area. Following that, we explain spherical image epipolar geometry and optical flow patterns in Section 3. In Section 4, we explain our proposed estimation methods followed by an explanation of how to create the virtual, rotation-less view in Section 5 along with selected frames showing its working. In Section 6, we show experimental evaluation of our proposed approaches with a motion capture system. Finally, we conclude the paper in Section 7.

2. Related Work

Most work with regards to creating a stable, rotation-less camera makes use of additional sensors to estimate the rotation of a camera. For example, [5, 6] make use of inertial measurement units (IMUs) to estimate and correct the rotation. However, such units often have large drift and it is well known that image-based approaches give lesser errors. Moreover, it is always difficult to synchronize the capture of information when using multiple sensors. Such methods cannot be universally applied to all systems without addition/change of hardware. Since our purpose is to process videos, it would be more suitable to use image information alone. Similar to our approach, [7] and [8] used spherical cameras along with expensive depth sensors for mapping textures to high quality point clouds to provide an immersive view. While these systems generate high quality 3D outputs, they are quite cumbersome to design and calibrate, apart from the cost of the equipment and the power requirements. In most cases, a simple visual feed would suffice.

There are also methods which involve a full structure from motion estimate [9, 10]. They use the estimated camera trajectory to fully stabilize the video feed in 3D. However, they can be considerably heavy in terms of the performance and very heavy on the memory as they always involve a global bundle adjustment step. Although they are globally consistent and accurate, such heavy methods are not usually required for stabilization purposes and neither are they feasible in scenarios where the robot moves across vast distances. Frame-by-frame approaches that track camera motion are usually sufficient for stabilization and do not involve intensive steps like the creation, handling, and storage of a 3D map of the environment. Such methods not involving 3D reconstructions are also available [11, 12], but they are based on sparse feature tracking and can easily be affected by the presence of many outliers. Limiting the estimation to use information contained in a few sparse points makes it susceptible to noise and lowers accuracy. Meanwhile, area-based approaches that work on optimizing a dense error function all over the image are quite precise for such estimates and very well suited for spherical images as they cover a very wide view. They are also stable because they make use of information present in every pixel of the image, thus removing outliers or reducing their effects. Considering these advantages, [13] introduced the use of the spherical Fourier transform to estimate rotations between two spherical images. However, their method fails to account for the effects of translation.

To characterize pixel motion, dense optical flow becomes the logical choice of information. Dense optical flow methods like [14] are very popular for perspective video stabilization. Hence, we attempt the same on spherical videos. Earlier, [15] made use of dense optical flow to estimate precisely the epipolar geometry of perspective images. They explained how the smoothing/regularization property of global dense optical flow methods helps in removing any local outliers and reducing their effects. In case of regularization/smoothing, outliers are reduced to Gaussian noise. Previously, [16] discussed the types of flow patterns formed on a spherical camera under pure rotation and translation. They explained how a spherical image frame in a video can be derotated to the same orientation as the frame before it based on optical flow in order to separate the translational and rotational components, which can give us the motion parameters of the camera. Thus, they followed a pattern recognition-like approach with multiple searches for the motion along 3 separate axes. This could be slow in practice and their work was left in a theoretical state. A similar approach was also followed in [17] for motion estimation, but using sparse flows. Meanwhile, [18] also followed a similar technique for stabilizing the roll and pitch motions of a mobile robot. However, their motion model was designed for ground robots moving in a plane.

In contrast to the above, our proposal is to use the entire dense optical flow information at every pixel of the image in order to estimate the rotation and stabilize the camera, without any assumption about the type of motion. Following this, the spherical video can be unwarped at an point to generate a rotation-less virtual camera. As we show, this usage of dense information results in a smooth stabilization with less errors as compared to sparse feature point information. We proposed a similar approach in an earlier paper [19] that estimated only the rotation in 3 DoF using a simple approach. In this paper, we explain it in detail and propose an extension to make it more accurate by including the full 5 DoF epipolar estimate using a different error function. We also evaluate the performance of both methods and compare them to estimation using sparse feature information alone to show the benefit of using dense information.

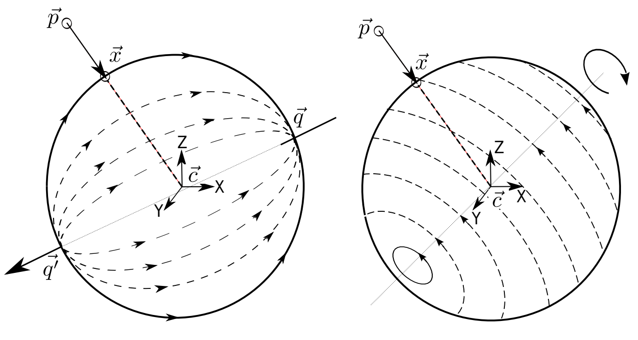
3. Spherical Epipolar Geometry and Optical Flow

In order to understand optical flow patterns on spherical images, an explanation of spherical image geometry is necessary. As the name suggests, spherical images are formed on a sphere, rather than a plane. Assuming a calibrated camera, every pixel $\vec{x}$ with Cartesian coordinates (X,Y,Z) forms a unit radius vector on the spherical image S. The center of the spherical image S is set as the origin. (Since it is difficult to show spherical images, we show all images in the equirectangular format.) As explained in [20], pixels on the spherical camera under motion also follow the same mathematical constraints under motion as any central projective camera. Hence, we can define:

$$\mathbf{x}'^{T}\mathbf{E}\mathbf{x} = 0 \quad \dots\dots\dots\dots\dots\dots \quad (1)$$

where $\mathbf{E}$ is the essential matrix between the two images and $\mathbf{x}$ and $\mathbf{x}'$ are the corresponding points of the same 3D environmental point projected on the two images, written with Cartesian coordinates in a column matrix form as $\begin{bmatrix} X & Y & Z \end{bmatrix}^{\top}$. It can be estimated using a regular 8-point RANSAC algorithm [21] and decomposed to give the rotation matrix and translation vector. Thus, the frame-to-frame motion of the camera can be completely defined in 5 DoF: 3 for the rotation, and 2 for the direction of translation. The frame-to-frame translation only requires 2 degrees of freedom as it has no magnitude defined. This is because cameras lose all information about distance. A relative global scale can be defined to obtain the camera trajectory, but is not necessary for our purpose of stabilization. In this research, we are only interested in finding the 3 DoF camera rotation to stabilize the video.

As for the physical interpretation of point movement on the unit sphere, [16] described how the image points move for the camera undergoing pure translation and pure rotation. These patterns are shown in **Fig. 2**. It can be seen that for pure translational movement, all points move on the unit sphere move along circles joining the two epipoles $\vec{q}$ and $\vec{q}'$. These are known as the epipolar circles, analogous to epipolar lines in planar images and they describe the path of pixels on the image for the camera undergoing translation. Meanwhile for pure rotation, the points move in circular loops around the axis of rotation.



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

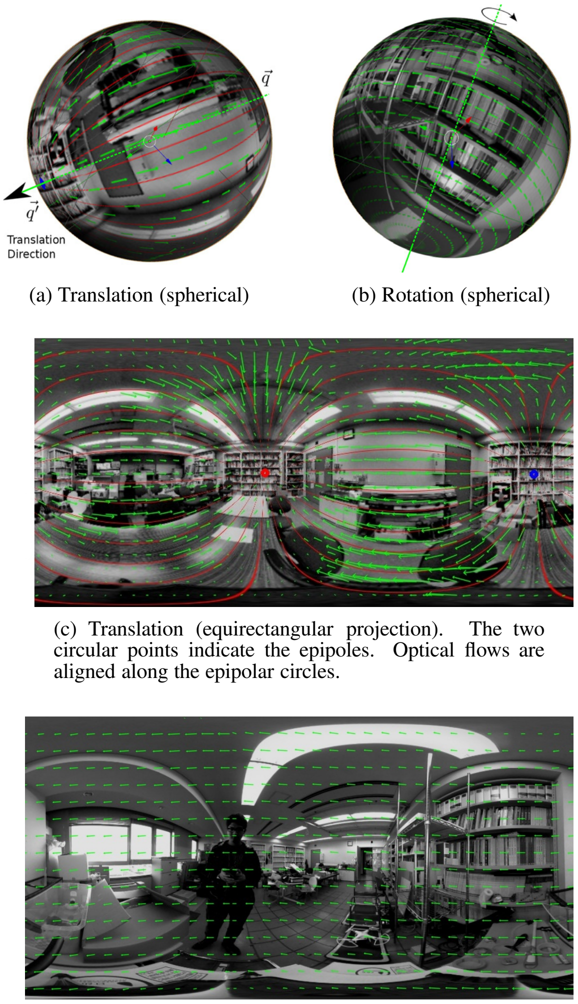
Fig. 2. Movement of pixels on the unit sphere for (a) pure translation (b) pure rotation. The arrows indicate the directions of translation and rotation, respectively.

In order to confirm these patterns, we calculated the optical flow on real situations of pure translation and rotation using the Farneback algorithm [22]. Images were captured with a spherical camera (Ricoh Theta S) under pure rotation and pure translation using a graded camera stand. The flow vectors induced by the images were calculated in the equirectangular projection[1] and projected on the unit sphere. The resultant flow vectors are seen in **Fig. 3**. The similarity with the patterns in **Fig. 2** can be clearly noticed.

It is obvious that the frame-to-frame arbitrary motion of the camera can be represented as a combination of rotation and translation. Thus, it follows that if we find the rotation between two spherical images and derotate one image to the same orientation as the other, the flow pattern between them will be purely translational in nature, as shown in **Figs. 2(a)** and **3(a)**. This removal of rotation is possible in spherical images because they encode information from all directions, as opposed to planar images.

Our proposed algorithm relies on this concept of derotation in order to find the rotation of the camera and stabilize it. We propose two dense approaches for this. One of them involves only the 3 DoF frame-to-frame rotation parameters and ignores translation using a unique symmetry property of fully spherical vision. The other involves the entire 5 DoF frame-to-frame epipolar estimate and is slower, but more accurate than the former. In the end, we are only interested in the 3 DoF rotation estimate for stabilization. To the best of our knowledge, no similar approach involving use of dense flow information has been proposed for spherical image rotation estimation. One work that came close, but used feature points for rectification was explained in [23]. Our proposed approach delivers a more accurate result as compared to using only sparse feature points, as shown later via experiments.

[1] In order to view spherical images in their entirety, they are displayed in the equirectangular format. All the processing mentioned in this research, however, is done on a spherical space.



(a) Translation (spherical)

(b) Rotation (spherical)

(c) Translation (equirectangular projection). The two circular points indicate the epipoles. Optical flows are aligned along the epipolar circles.

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

Fig. 3. Estimated motion fields on the spherical image for translation and rotation in both spherical and equirectangular projections. Patterns similar to **Fig. 2** above can be noticed. (Note: the flow was calculated densely, but for visualization, it been shown sparsely.)

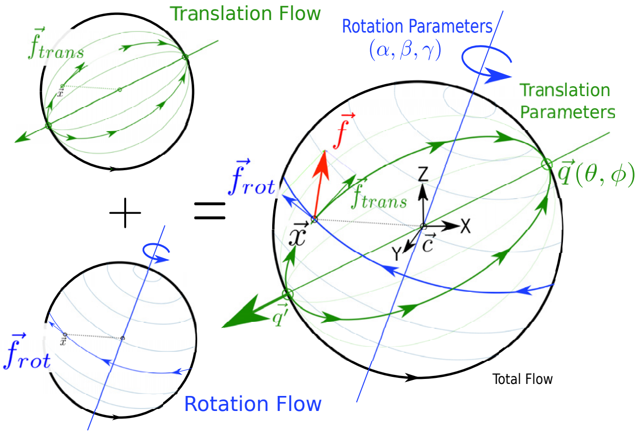


Fig. 4. Motion model and notations. All translation parameters are represented by epipole $\vec{q}(\theta,\phi)$. All rotation parameters are represented by the $<X$-Y-$Z>$ Euler angles α, β, and γ. $\vec{f}_{trans}$ is the component of optical flow caused by translation while $\vec{f}_{rot}$ is the component due to rotation. The total optical flow vector $\vec{f}$ is $\vec{f}_{trans} + \vec{f}_{rot}$.

4. Rotation Estimation by Derotation

With no assumptions about the camera trajectory, we track the frame-by-frame camera rotation and stabilize the video to generate a rotation-less output. It is obvious that any movement of the camera between two frames involves a rotation and translation. If we perfectly estimate the rotation of the current frame from the previous frame and reverse it, we should end up with a purely translational alignment. Thus, if we define dense measures of similarity to a translational state, we should be able to optimize the rotation of one image frame with respect to the previous one and remove it. We propose using the dense optical flow pattern to define this similarity measure using two different approaches – a simple one based on the moment of optical flow vectors that can ignore the effect of translation, and another based on the full epipolar constraint including the direction of translation. The basic idea is to formulate a minimization algorithm in order to bring the dense optical flow pattern to that of a translational pattern in **Figs. 2(a)** and **3(a)** in order to estimate the rotation. First, we will explain the mathematical notations and the motion model we follow on the unit sphere.

4.1. Notation and Motion Model

We define rotation using the Euler angles α, β, and γ in the X-Y-Z axes notation. The main reason for using Euler angles is to ensure a small search space. Typically, the frame-to-frame rotations are small. Hence, all parameters will remain close to zero. Every pixel is denoted as its radius unit vector $\vec{x}$ in Cartesian coordinates. As for the translation parameters, we indicate it by the one of the epipoles $\vec{q}$ (the point opposite to direction of motion of the sphere). Since $\vec{q}$ has to lie on the surface of a sphere, it is defined in terms of the spherical coordinates θ and ϕ only for the minimizations. This implicitly enforces the spherical condition and ensures that all parameters take the form of angles. Thus, $(\alpha, \beta, \gamma, \theta, \phi)$ uniquely define the 5 DoF motion between two spherical images.

The optical flow vector at $\vec{x}$ is written as $\vec{f}$, relative to $\vec{x}$. $\vec{f}_{trans}$ is defined as the translational component of optical flow, which is tangential to the epipolar circle connecting $\vec{x}$ and the epipole $\vec{q}$. $\vec{f}_{rot}$ is the component caused by rotation, tangential to the circle passing through $\vec{x}$, whose plane is perpendicular to the axis of rotation. All parameters and notations can be seen in **Fig. 4**.

4.2. Derotation by Minimization

We describe two different approaches to estimate the rotation by minimizing the deviation from a pure translational state. The first is a simpler function that estimates rotation by ignoring translation, using a unique property of complete spherical imaging.

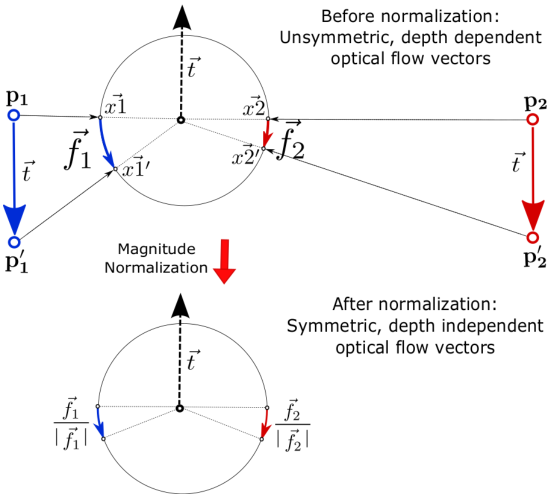


Fig. 5. Diametrically opposite points on the sphere induce moments in opposing directions for pure translation. Normalizing the magnitudes of optical flow vectors removes depth dependency from the pairs of points and makes it, and by extension the entire optical flow field, completely symmetric about the center.

4.2.1. 3 DoF Directional Moment Approach

In a pure translational movement, all image points move along epipolar circles joining the two epipoles. Every pixel formed from a 3D point in the environment has a diametrically opposite point moving in the same direction, as seen in **Fig. 5**. These diametrically opposite pairs of points induce moments around the center of the sphere in opposite directions. Thus, a *directionally symmetric* optical flow pattern is formed, as seen in **Figs. 2(a)** and **3(a)**. The magnitude of the optical flow vector at each pixel depends on its distance from the camera. However, our objective is not to find this depth, but only the motion of the camera. Hence, to remove the depth dependence, we divide all optical flow vectors by their magnitude, (or, in other words, discard the magnitude of the vector and retain the direction of the vector). Then, every diametrically opposite pair, and by consequence, the entire pattern becomes completely symmetric. An example of this is shown in **Fig. 5**. The camera is moving upwards and two points $\mathbf{p_1}$ and $\mathbf{p_2}$ are at diametrically opposite, different distances from the center. It can be seen that the optical flow induced by $\mathbf{p_1}$ is less than that of $\mathbf{p_2}$. However, dividing each by its magnitude preserves the direction information and induces symmetry in both. The same can be said of all diametrically opposite point pairs.

In this state, independent of the translation direction, it can be said that the 'directional moment' of the optical flow vectors around the center of the camera should be zero. We exploit this property of the fully spherical optical flow field to find its rotation. Thus, we define the following quantity, known as the 'directional moment' of the optical flow field:

$$\vec{M} = \sum_{\forall \vec{x} \in S} \left(\vec{x} \times \frac{\vec{f}}{|\vec{f}|} \right). \quad \ldots\ldots\ldots\ldots \quad (2)$$

This quantity can be calculated entirely from the state of the optical flow at a frame. If we rotate one of the images by some angles α, β, and γ, we may reach a point at which this quantity reaches zero. In this state, the frame can be said to have no rotation with the previous frame. With this in mind, we define a minimization of the directional moment of optical flow with respect to the rotation angles α, β, and γ in order to estimate them:

$$\underset{\alpha,\beta,\gamma}{\text{minimize}} |\vec{M}| = \underset{\alpha,\beta,\gamma}{\text{minimize}} \left| \sum_{\forall \vec{x} \in S} \left(\vec{x} \times \frac{\vec{f}}{|\vec{f}|} \right) \right|. \quad . \quad (3)$$

Any deviation from the zero moment means that a rotational flow is induced on the spherical image. Thus, the directional moment $\vec{M}$ can be used as a measure of 'deviation' from the translational state. It can be calculated as follows:

- 'De-rotate' the frame with Euler angles α, β, and γ along the x-y-z Euler axes.

- Calculate the optical flow vector $\vec{f}$ at every point $\vec{x}$ with respect to the previous frame.

- Normalize the value of each flow vector $\vec{f}$ as $\frac{\vec{f}}{|\vec{f}|}$.

- The total error is computed as the magnitude of the directional moment $\vec{M}$ (Eq. (2)).

The directional moment can be calculated using only the 3 rotation angles and is thus in 3 DoF. The error or the measure of deviation from a translational state can be estimated without involving the translation by using the unique property that the field of view is completely spherical and hence directionally symmetric in a purely translational state. This symmetry is not possible for a perspective camera which has information only from a narrow field of view and thus, this approach is unique to a spherical camera. It gives suitably good performance in indoor areas with rich textures. Due to the requirements of symmetry, the inclusion of the robot body inside the image induces problems. This can easily be rectified by simply excluding the part of the image where it appears as well as the diametrically opposite region. However, the assumption of good estimation of optical flow direction from all around the camera may lead to sub-optimal solutions and increase drift in case of environmental regions with many missing textures where the optical flow direction is ambiguous. Although this can usually taken care of to an extent due to the regularization property of optical flow, we introduce our next error function, based on the full frame-to-frame epipolar constraint rather than symmetry conditions. We expect it to be slightly more accurate in comparison to this error function.

4.2.2. 5 DoF Direct Epipolar Approach

Normally, frame-to-frame camera motion is expressed in 5 DoF. 3 for rotation, and 2 for translation direction.

However, in the previous subsection, the proposed minimization assumed a symmetry condition in the translational flow field and thus was able to restrict the estimation to the 3 DoF rotation, independent of the translation direction. It could lead to a sub-optimal solution in cases where the optical flow direction cannot be determined properly in every direction. Now, we propose another minimization – one based directly on the full 5 DoF frame-to-frame epipolar constraint. Instead of assuming a symmetric flow with an unknown translation direction, we include the translation direction in the optimization and try to align all optical flow vectors in accordance to it. It is an approach similar to non-linear epipolar estimation algorithms, but one that takes advantage of dense information and regularization imposed by dense optical flow algorithms, as explained in [15].

In the same manner as above, we define an error function to indicate deviation from a translational state of motion, this time based on the full epipolar constraint. A pure translational state requires that all the flow vectors (pixel movements) are pointing away from the direction of translation. In other words, they should be pointing towards epipole $\vec{q}$, as shown in **Figs. 2(a)** and **3(a)**. At every point $\vec{x}$, optical flow vector (pixel movement) should point towards the epipole $\vec{q}$, tangential to its respective *epipolar circle* which joins $\vec{x}$ to $\vec{q}$. Any deviation will cause this arrangement to be disturbed. As suggested in [24] which used an angular error for the epipolar constraint (using sparse feature points), we also use a similar error for optical flow.

Thus, our error function is defined over the rotation angles α, β, and γ, and the translation direction, which is opposite to epipole $\vec{q}(\theta,\phi)$. Assuming perfect estimation of all parameters, each point $\vec{x}$ on the sphere should have its current optical flow vector $\vec{f}$ tangential to its epipolar circle, i.e., the circle connecting $\vec{x}$ to $\vec{q}(\theta,\phi)$. In other words, $\vec{f}_{rot}=0$ and $\vec{f}=\vec{f}_{trans}$. Therefore, we can define the difference between $\vec{f}$ and $\vec{f}_{trans}$ as the error at point $\vec{x}$. $\vec{f}_{trans}$ depends on the 3D structure of the environment. However, its direction can be estimated from the property that it is aligned tangential to the epipolar circle connecting $\vec{x}$ to $\vec{q}$ $(\theta,\ \phi)$. Thus, we consider the angle between $\vec{f}$ and $\vec{f}_{trans}$ as the error at point $\mathbf{x}$ denoted by Ω. Ω can be calculated as follows (as shown in **Fig. 6**):

We find the epipolar circle C_q (as shown in **Fig. 6**) as defined by its normal vector $\vec{N}_q$:

$$\vec{N}_q=\vec{q}\times\vec{x}. \qquad (4)$$

Similarly, we can find another circle C_f (as shown in **Fig. 6**) to which $\vec{f}$ is tangential, by its normal vector $\vec{N}_f$:

$$\vec{N}_f=(\vec{x}+\vec{f})\times\vec{x}. \qquad (5)$$

Now, the angle Ω, which forms the error at pixel $\vec{x}$ is equal to the angle between $\vec{N}_q$ and $\vec{N}_f$:

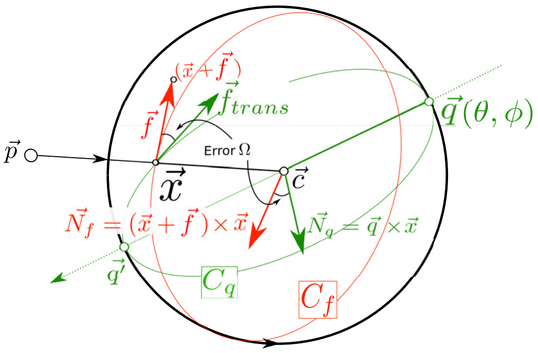


Fig. 6. Calculating the error at $\vec{x}$. To find angle Ω, we take the cross products $(\vec{x}+\vec{f})\times\vec{x}$ and $\vec{q}\times\vec{x}$ in order to define the circles along $\vec{f}_{trans}$ and $\vec{f}$ and find the angle between them.

$$\Omega=\arccos\left(\frac{\vec{N}_q\cdot\vec{N}_f}{|\vec{N}_q||\vec{N}_f|}\right) \qquad (6)$$

$$=\arccos\left(\frac{(\vec{q}\times\vec{x})\cdot\left((\vec{x}+\vec{f})\times\vec{x}\right)}{|(\vec{q}\times\vec{x})|\left|\left((\vec{x}+\vec{f})\times\vec{x}\right)\right|}\right) \qquad (7)$$

where $\vec{f}$ is calculated on derotating the image with the rotation parameters (α,β,γ), and $\vec{q}$, the epipole, is the Cartesian coordinate transform of (θ,ϕ). The total deviation from a translational state is defined as a least squares error of Ω over all pixels $\vec{x}$ on the spherical image.

To summarize the error formulation in each iteration for the 5 DoF parameters $(\alpha,\beta,\gamma,\theta,\phi)$:

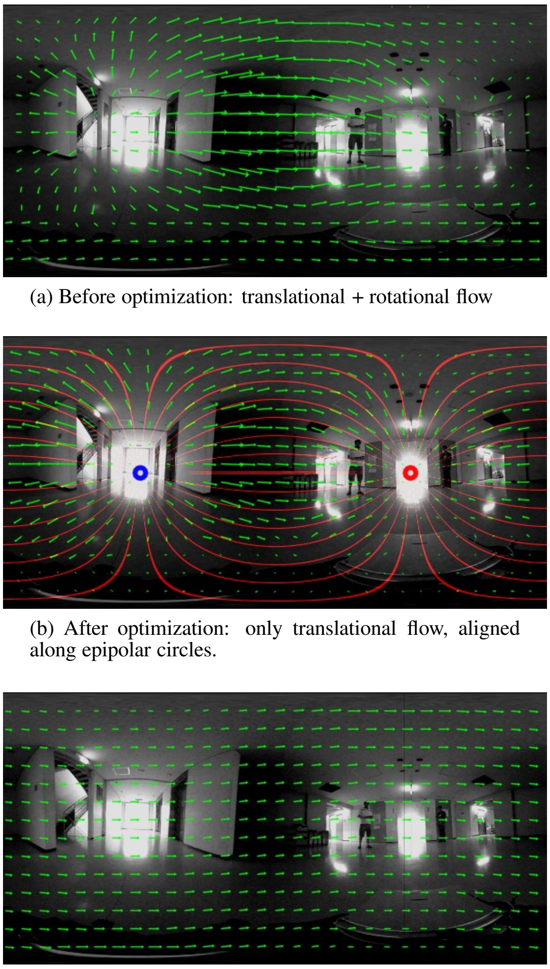
- ‘De-rotate’ the image with Euler angles α, β, and γ along the *x*-*y*-*z* Euler axes.
- Calculate the optical flow vector $\vec{f}$ at every point $\vec{x}$ with respect to the previous frame.
- According to the current estimate of epipole $\vec{q}(\theta,\phi)$, calculate for all $\vec{x}$ the angle between $\vec{f}_{trans}$ and $\vec{f}$.
- The error at point $\vec{x}$ is this angle, Ω (calculated from normal vectors $\vec{N}_f$ and $\vec{N}_q$).

The final optimization is posed as follows:

$$\underset{(\alpha,\beta,\gamma,\theta,\phi)}{\text{minimize}}\sum_{\forall\vec{x}\in S}\Omega^2. \qquad (8)$$

4.3. Minimization

Both approaches use different error functions to bring the frame-to-frame camera motion to a translational state. The directional moment approach is defined over the rotation angles α, β, and γ, and the direct epipolar approach is defined over those as well as the translation direction $\vec{q}(\theta,\phi)$. We employ the Levenberg-Marqardt [25] approach to solve the minimization problem for both approaches. Since it is difficult to define a discrete coordinate space on a sphere, we sum the error over all pixels of the equirectangular images, normalizing for the distortion induced on it in the same manner as [26].



(a) Before optimization: translational + rotational flow

(b) After optimization: only translational flow, aligned along epipolar circles.

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

Fig. 7. Spherical flow field (a) before optimization: translation + rotation. (b) After optimization: translation flow aligned along epipolar circles (the circles are distorted because they are shown in the equirectangular projection). The estimate of $\vec{q}$ is shown as the circular dot on the right side. (c) After optimization: (a)–(b) rotational flow (similar to **Fig. 3**).

For each consecutive pair of frames, both approaches output the rotation angles α, β, and γ for which the optical flow vectors appear to be purely translational, using either the symmetry property (directional moment approach) or with a full epipolar minimization (direct epipolar approach). **Fig. 7** shows an example of this minimization in one frame. Before optimization, the optical flow field is a mix of rotation and translation and the optical flow vectors are unaligned. After optimization, all the vectors are aligned along the epipolar circles. Moreover, subtracting the final optical field from the initial field reveals a rotation like-optical flow pattern, confirming that there was convergence. The result for this image was computed using the direct epipolar approach to show the epipole location. The computation using the directional moment error is not shown because there was no visually noticeable difference.

Since the direct epipolar approach involves a complicated non-linear least squares error, it is important to confirm that is is suitable for such a minimization. In order to do this, we decided to plot the error function Ω over a range of values of rotation and translation. The translation vector and rotation angles were each varied along arbitrary axes in space. The same pair of images (same as the ones used in **Fig. 7**) were used for this computation. **Fig. 8** shows the value of the least squares direct epipolar error Ω plotted with respect to the translation and rotation errors on these images. It can be seen that the convex shape of this function with a single minimum is suitable for a minimization.

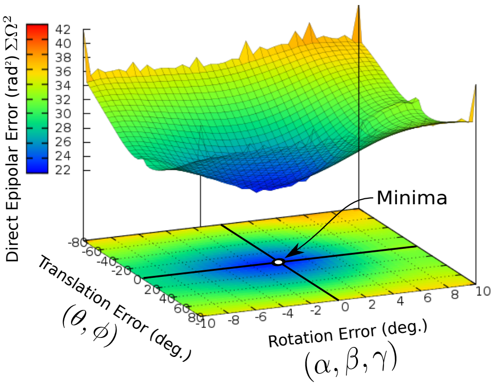


Fig. 8. Direct epipolar error against translation (epipole) and rotation errors.

Thus, two different approaches can be used for rotation estimation. Both are used to track the rotation in a frame-by-frame manner. The directional moment approach is a little less accurate in case of many textureless regions where optical flow direction cannot be suitably determined. To a large extent, textureless regions are taken care of by most dense optical flow approaches that employ any sort of spatial regularization. It is a 3 DoF optimization which is faster than calculating the full epipolar geometry and is enough for stabilization. Meanwhile, the direct epipolar error is based on all 5 DoF of the epipolar geometry. It does not make use of the symmetry condition and instead involves the epipolar direction in the optimization, directly optimizing the dense pixel movements along epipolar lines. However, being a 5 DoF optimization, it is slower than the directional moment error.

4.4. Reprojection of Optical Flow

Both proposed approaches involve derotation based minimization. The error function or deviation from a translational state can only be evaluated after derotating the frame with estimates of α, β, and γ and checking the optical flow state of the image each time. Although it would be more accurate to re-estimate optical flow vectors in each iteration, it could make the algorithm quite heavy. Instead, within a reasonable limit, the optical flow vectors can simply be reprojected based on the initial state. The resulting flow will be close enough to the estimated optical flow at that state. Given the initial optical flow vector $\vec{f_i}$ at $\vec{x}$, our objective is to calculate the optical flow vector $\vec{f}$ at any estimate of the rotation angles α, β, γ. $\vec{x}$

lies on one frame (the one that is rotated) and $\vec{x}+\vec{f_i}$ is its corresponding point on the next frame (that remains unchanged). Thus, we rotate $\vec{x}$ with the rotation matrix R (composed as $R_x(\alpha)R_y(\beta)R_z(\gamma)$) as shown in Eq. (9):

$$\vec{x_r} = R_x(\alpha)R_y(\beta)R_z(\gamma) \times \vec{x}. \quad \ldots\ldots\ldots \quad (9)$$

Following this, the optical flow vector can be calculated by subtracting the new point $\vec{x_r}$ from $\vec{x}+\vec{f_i}$:

$$\vec{f} = (\vec{x}+\vec{f_i}) - \vec{x_r}. \quad \ldots\ldots\ldots\ldots \quad (10)$$

4.5. Initial Estimate

4.5.1. Directional Moment Approach

In the directional moment error approach, the parameter vector consists only of the rotation angles α, β, and γ. Since we follow a frame-by-frame approach, the rotations from one frame to another are typically small (less than 10° in total, as noticed). Hence, for each frame, the initial value can be simply set to zero.

4.5.2. Direct Epipolar Approach

The direct epipolar error is a non-linear least squares problem and hence requires a good initial value for the epipole $\vec{q}(\phi,\theta)$ in addition to the rotation angles α, β, and γ. It is difficult to find a suitable initial value for the translation as it can be in any arbitrary direction in space. Thus, for this approach, we used an initial estimate from sparse feature points and refined it with the direct epipolar error.

4.6. 8-Point Approach and Refinement

Using an initial value of zero rotation is applicable only when optical flow can be easily computed from frame to frame. In case of rapid camera movement, the orientations of consecutive frames can significantly change, making optical flow computation infeasible. Moreover, the direct epipolar approach requires an initial value of the translation as well. In such cases, sparse feature point matching with the 8-point algorithm [21] must be used to provide a suitable starting point. It is essentially a linear solution of the epipolar constraint equation:

$$\mathbf{x}'^{T}\mathbf{E}\mathbf{x} = 0 \quad \ldots\ldots\ldots\ldots\ldots \quad (11)$$

where $\mathbf{E}$ is the essential matrix between the two images and $\mathbf{x}$ and $\mathbf{x}'$ are the corresponding points in the two images written in column matrix form as $\begin{bmatrix} X\ Y\ Z \end{bmatrix}^{\top}$. In our approach, we use A-KAZE features [27] to do find an initial point matching as it is particularly suitable for highly distorted equirectangular frames. RANSAC [4] is used to eliminate outliers, and a linear fit of all the inliers is chosen as an estimate for $\mathbf{E}$. Following this, $\mathbf{E}$ is decomposed into the rotation matrix and translation vector via Singular Value Decomposition (SVD). The estimated rotation matrix is converted to the three <*X*-*Y*-*Z*> Euler angles α, β, and γ and the translation vector is normalized for magnitude and converted to spherical coordinates θ and ϕ. Following this, our proposed derotation based minimization is used to refine the estimate.

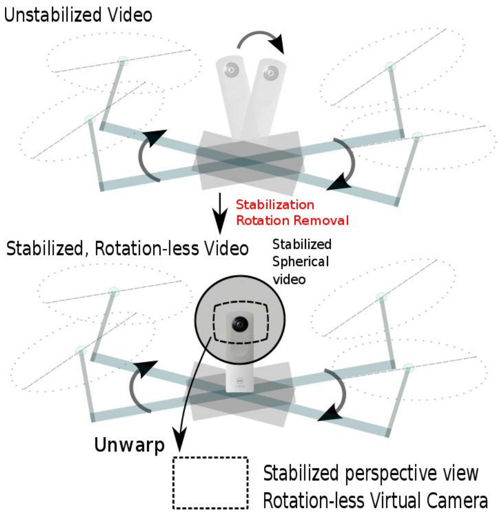

Unstabilized Video
Stabilization
Rotation Removal
Stabilized, Rotation-less Video
Stabilized
Spherical
video
Unwarp
Stabilized perspective view
Rotation-less Virtual Camera

Fig. 9. Pipeline for creation of a virtual rotation-less camera.

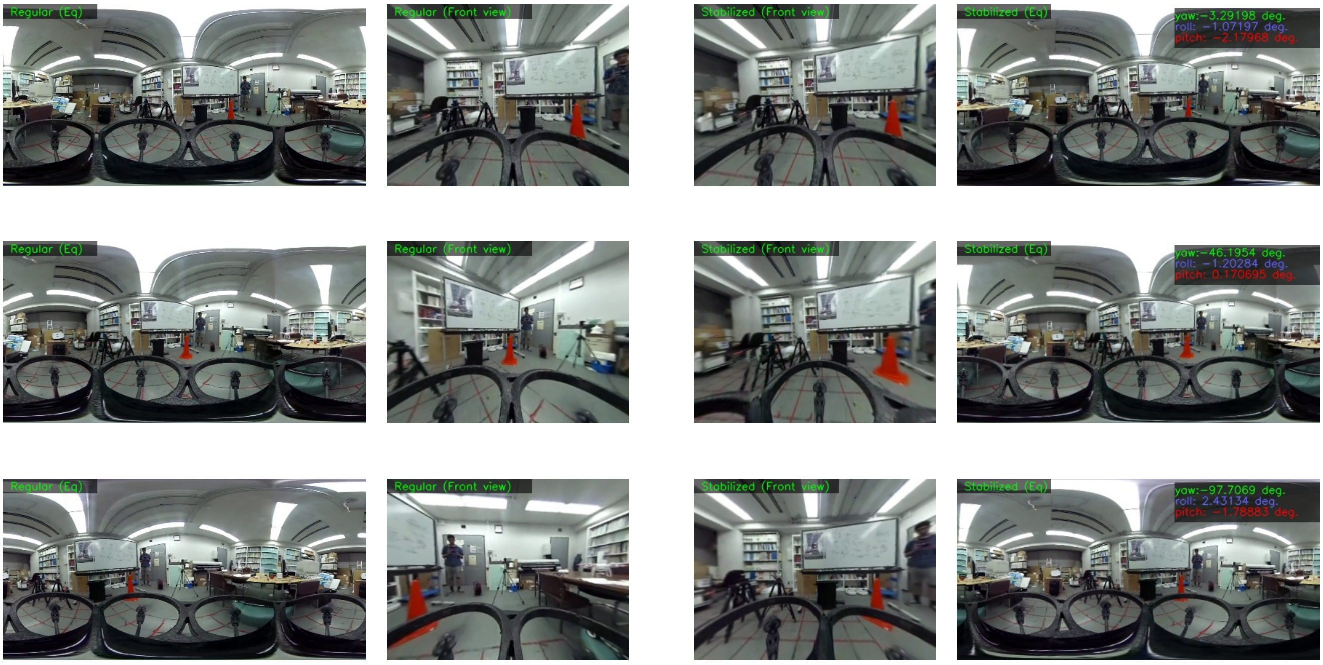
5. Video Stabilization and Virtual Rotation-Less Viewpoint

Using the proposed motion estimation, we can derotate each frame in a spherical video to finally end up with a stabilized, rotation-less video. Following this, any region on the spherical video can be unwarped to a perspective view. This forms the virtual rotation-less camera, as shown in **Fig. 9**.

In order to demonstrate this, videos were recorded by moving the spherical camera in various ways – by placing it on an AR Drone 2.0, a Pioneer P3-DX robot, and then by hand. The resultant videos were stabilized using both approaches. Since there was not much visual difference in the results from both approaches, we decided to show the more accurate of the two – the direct epipolar approach. The unwarped perspective view, which can be oriented as desired, forms the virtual rotation-less camera.

As for the results, series of 3 frames from each resultant video[2] are shown in **Figs. 10**, **11**, **12**, and **13** in order to demonstrate the effect of stabilization. In each case, the unstabilized frames (equirectangular and perspective) are shown next to the stabilized ones. The perspective stabilized frames form the output of the virtual rotation-less camera. It can be seen that the motion in the unstabilized frames is haywire and it is difficult to concentrate on the image. Whereas, in the stabilized case, the orientation remains the same making the the environment and camera displacement more easily perceptible. This orientation can be changed as desired while viewing the video. This is a unique way to view a spherical video and has applications in robotics for surveying, inspection, etc. as well as leisure and entertainment using spherical cameras. Next, we evaluate the performance of both approaches and compare them to a similar estimation done using sparse feature points.

[2] A link for all the videos is given at the end of the paper [a].



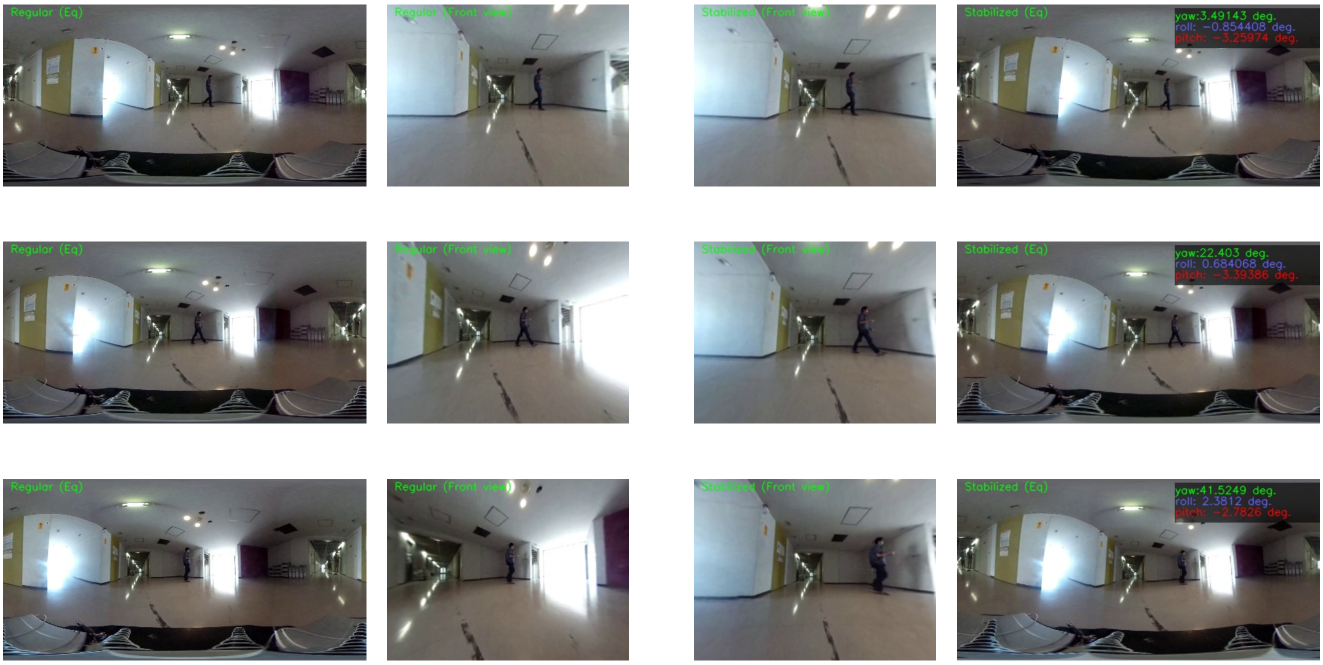
Unstabilized equirectangular

Unstabilized perspective

Virtual rotation-less (stabilized perspective)

Stabilized equirectangular

Fig. 10. Frames 30, 68, and 80 (top to bottom) from the AR Drone 2.0 sequence: equirectangular view and perspective undistorted front, regular and stabilized. The fixed orientation in the virtual rotation-less camera can be noticed.



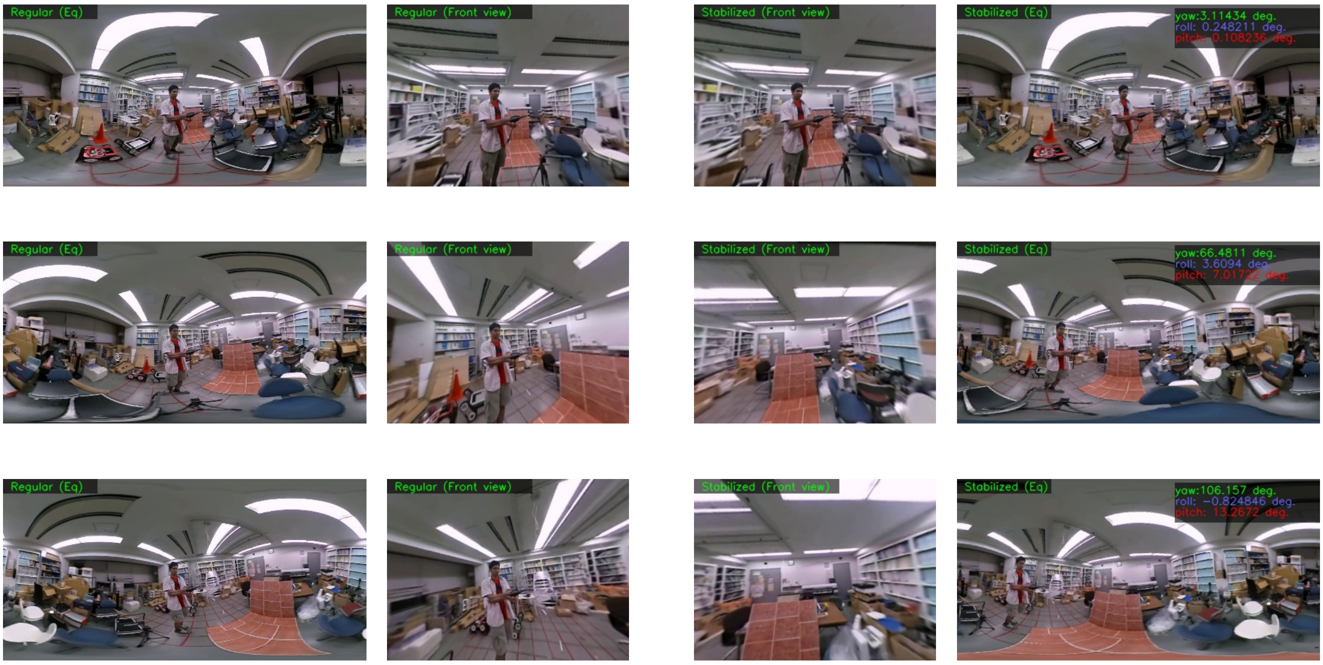
Unstabilized equirectangular

Unstabilized perspective

Virtual rotation-less (stabilized perspective)

Stabilized equirectangular

Fig. 11. Frames 100, 105, and 110 (top to bottom) from the Pioneer robot sequence: equirectangular view and perspective undistorted front, regular and stabilized. The fixed orientation in the virtual rotation-less camera can be noticed.



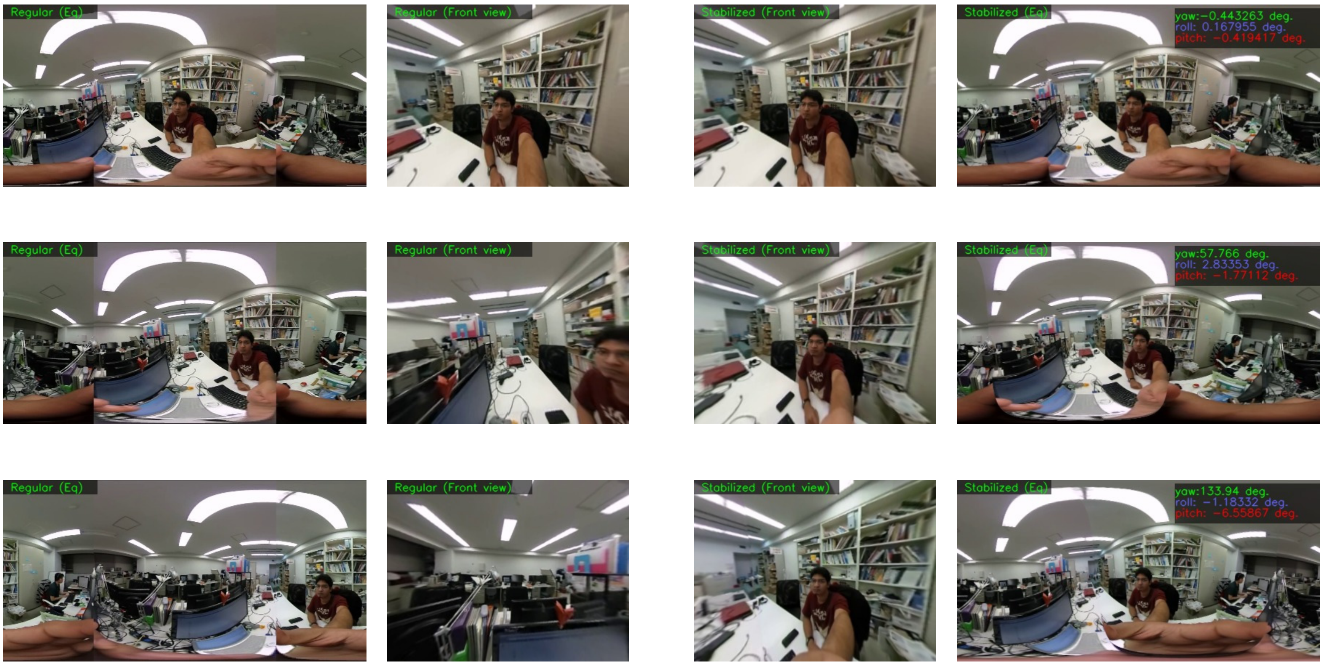
Unstabilized equirectangular

Unstabilized perspective

Virtual rotation-less (stabilized perspective)

Stabilized equirectangular

Fig. 12. Frames 1, 30, and 45 (top to bottom) recorded by moving the spherical camera by hand: equirectangular view and perspective undistorted front, regular and stabilized. The fixed orientation in the virtual rotation-less camera can be noticed.



Unstabilized equirectangular

Unstabilized perspective

Virtual rotation-less (stabilized perspective)

Stabilized equirectangular

Fig. 13. Frames 1, 50, and 80 (top to bottom) recorded by moving the spherical camera by hand: equirectangular view and perspective undistorted front, regular and stabilized. The fixed orientation in the virtual rotation-less camera can be noticed.

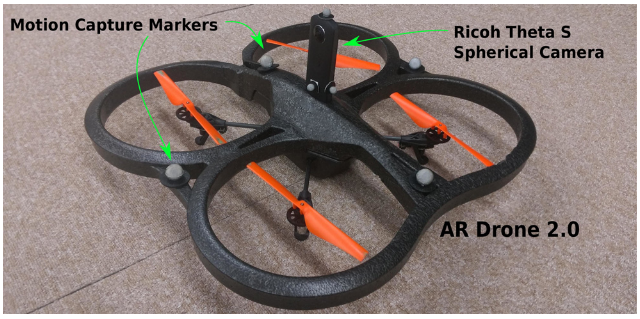


Fig. 14. Experimental setup: AR Drone with MoCap markers and the mounted Ricoh Theta S spherical camera.

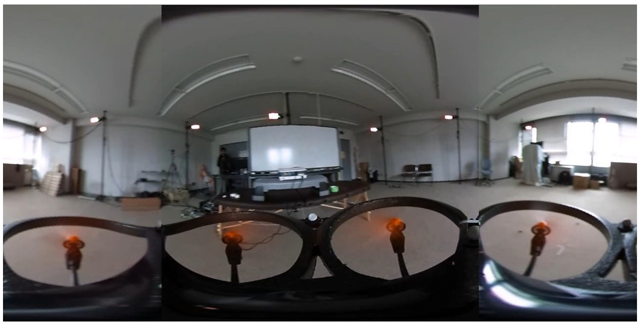


Fig. 15. One frame from the evaluation experiment.

6. Evaluation

6.1. Experimental Setup

The previous section explained and demonstrated the generation of a virtual rotation-less view. In this section, we evaluate and quantify the performance of both approaches described and compare them to estimation using sparse feature information alone.

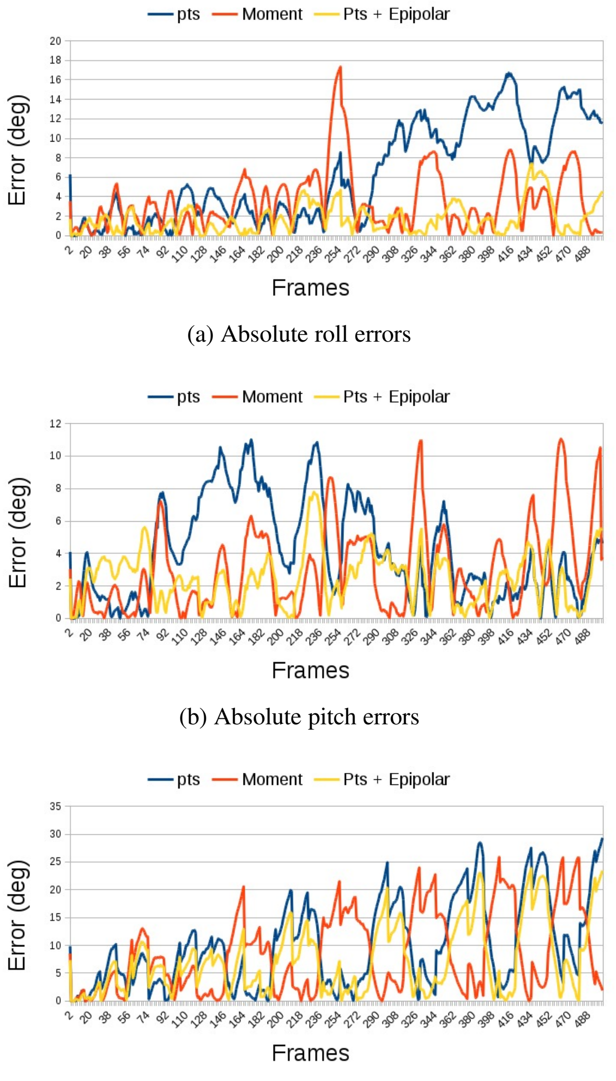
A millimeter precise motion capture system was used in order to record groundtruth rotation information to test the accuracy of the algorithm to estimate the motion. An AR Drone 2.0 was fitted with the Ricoh Theta S camera and several optical markers. It was flown in a room equipped with a Cortex *MotionAnalysis* setup. A video was recorded at 30 frames per second. The roll, pitch, and yaw motions were calculated with respect to the orientation in the first frame and taken to be the groundtruth. Since the motion capture system had a recording rate of 100 frames per second, the rotation values obtained were interpolated and converted to a 30 FPS rate in order to match it with the recording rate of the camera. The start of the movement in both the motion capture system and the recorded video were synchronized manually. The setup is shown in **Fig. 14**. One example frame from the evaluation experiment is shown in **Fig. 15**. Around 500 such frames were captured and processed.

The drone underwent complex motions of yaw (complete 360°), roll (from -25° to $+15^\circ$), and pitch (from -15° to $+15^\circ$). The rotation angles were calculated from the video in three ways and compared to the groundtruth: using only sparse feature points with 8-pt RANSAC [21] (*pts*), and using our proposed dense approaches – the directional moment approach (*Moment*), and using an initial value from sparse feature points (*pts*) and refinement with the direct epipolar approach (*pts* + *Epipolar*). As the directional moment error did not require any initial value, it was tested standalone to see its performance in relation to using sparse information. The recently developed A-KAZE features [27] were chosen to provide the sparse point information. A-KAZE provides several times greater performance as compared to any other descriptor on spherical images. These points were filtered in a 8-point RANSAC approach resulting in the best possible output of sparse information to be compared with. The Farneback optical flow [22] was used to estimate a smooth, dense flow field for all the results given in this paper. The reason for this choice was because it is an easy and fast approach and the amount of smoothing/regularization can be explicitly controlled by changing the regularization window.

6.2. Results

The absolute errors with each approach are shown in **Fig. 16** and the averages of all absolute errors are shown in **Table 1**. The results show that the dense optical flow based methods (‘Moment’ and ‘pts + Epipolar’) are more accurate as compared to using sparse feature points (‘pts’) alone. As expected, the direct epipolar approach (‘pts + Epipolar’) is a bit more accurate as compared to the directional moment approach (‘Moment’) which assumes accurate estimation of optical flow direction from all pixels. These results demonstrate the superiority of using dense information from all pixels as opposed to information only from specific feature points. Two issues which arise while using sparse feature information – outlier errors and information bias are both solved when using the dense optical flow field. The processing time is a bit slow – around 1 second per frame for the directional moment error (‘Moment’) and around 5 seconds per frame for the direct epipolar error (‘pts + Epipolar’) on a 2.8 GHz CPU without any parallel processing. However, the results are more accurate and especially suited for such video stabilization approaches.

The results also show other interesting properties. The errors for all three rise and fall in an alternate manner, especially for the yaw angle. This is because during the experiment, the drone underwent rapid changes along the yaw from right to left. Considering that our approach relies on a frame-by-frame estimation, the drift error is expected to accumulate and then reduce with motion in the opposing direction. Another important property of our approach can be noticed. The directional moment approach does not require any initial value due to its small search space. Hence, it exhibits error patterns different from the other two approaches. Meanwhile, the direct epipolar approach uses sparse feature points (‘pts’) as an initial value and reduced the error from that point onwards. Hence, while it exhibits lower estimation errors, the pattern of its



(a) Absolute roll errors

(b) Absolute pitch errors

(c) Absolute yaw errors

Fig. 16. Comparison of absolute roll, pitch, and yaw errors using sparse information and various proposed dense optical flow approaches. ('pts' – sparse feature points only, 'Moment' – directional moment approach, and 'pts + Epipolar' – initial value using sparse feature points and refined estimation using the direct epipolar approach.)

Table 1. Average absolute errors (all values in deg., lowest values in bold).

Approach	Average Abs. Errors		
	Pitch	Yaw	Roll
pts	4.09	9.84	6.29
Moment	3.05	8.53	3.54
pts + Epipolar	**2.44**	**7.36**	**1.72**

errors follows that of 'pts.' Due to the restriction of using the indoor motion capture system, the experiment could only be performed in a single environment. The effect of the environment on the estimation remains as future work.

7. Conclusions

In this paper, we proposed a method of using dense optical flow information to estimate the rotation of spherical videos and stabilize them to generate a virtual rotation-less camera. We explained two different approaches that make it possible to estimate the frame-by-frame rotation. The directional symmetry approach works in 3 DoF using a property of symmetry in optical flow directions on a spherical camera. It is interesting to note that this cannot be applied to a perspective camera and is unique to a spherical camera as it needs estimation of optical flow from diametrically opposite directions. The second is the direct epipolar approach that works in 5 DoF. It can be applied to perspective cameras as well, but its effectiveness is greatly enhanced when used with a spherical camera. It is much slower (around 5 times) as compared to the 3 DoF directional symmetry approach, but a bit more precise in its estimation as it is not affected as much by regions of missing optical flow estimation. We also described our approach in creating a virtual rotation-less camera for a spherical video. In this regard, the spherical camera essentially serves two purposes: providing the optical flow information from all directions for estimating the rotation, as well as color information from all directions to provide the view in that direction.

Using a motion capture system as groundtruth, we demonstrated lesser errors as compared to using sparse feature information alone, which can be affected by noise. Drift errors are inevitable in any approach based on incremental tracking and SLAM approaches attempt to solve this by building a global map. In this work, we focused on estimation without resorting to map building (which is an intensive, resource consuming problem) to satisfy the purpose of video stabilization. However, our proposed approach can be used in conjunction with global localization and SLAM techniques to increase accuracy when using a spherical camera. That, as well as a suitable translation scale estimation to establish an accurate visual odometry approach will be our future work.

Acknowledgements

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] S. Hughes, J. Manojlovich, M. Lewis, and J. Gennari, "Camera control and decoupled motion for teleoperation," Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics, pp. 1339-1344, October 2003.

[2] S. Hughes and M. Lewis, "Robotic camera control for remote exploration," Proc. of the SIGCHI Conf. on Human factors in Computing Systems, pp. 511-517, July 2004.

[3] D. G Lowe, "Distinctive image features from scale-invariant keypoints," Int. J of computer vision, Vol.60, No.2, pp. 91-110, January 2004.

[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Communications of the ACM, Vol.24, No.6, pp. 381-395, June 1981.

[5] T. Albrecht, T. Tan, G.A.W. West, and T. Ly, "Omnidirectional video stabilisation on a virtual camera using sensor fusion," Proc. of the Int. Conf. on Control Automation Robotics Vision, pp. 2067-2072, December 2010.

[6] R. Miyauchi, N. Shiroma, and F. Matsuno, "Development of omnidirectional image stabilization system using camera posture information," Proc. of the IEEE Int. Conf. on Robotics and Biomimetics, pp. 920-925, December 2007.

[7] K. Kruckel, F. Nolden, A. Ferrein, and I. Scholl, "Intuitive visual teleoperation for ugvs using free-look augmented reality displays," Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 4412-4417, May 2015.

[8] F. Okura, Y. Ueda, T. Sato, and N. Yokoya, "Teleoperation of mobile robots by generating augmented free-viewpoint images," Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 665-671, November 2013.

[9] M. Kamali, A. Banno, J. C. Bazin, I. S. Kweon, and K. Ikeuchi, "Stabilizing omnidirectional videos using 3d structure and spherical image warping," Proc. of the IAPR Conf. on Machine Vision Applications, pp. 177-180, June 2011.

[10] A. Torii, M. Havlena, and T. Pajdla, "Omnidirectional image stabilization by computing camera trajectory," Advances in Image and Video Technology, Springer Berlin Heidelberg, Vol.5414, pp. 71-82, January 2009.

[11] M. Kamali, S. Ono, and K. Ikeuchi, "An efficient method for detecting and stabilizing shaky parts of videos in vehicle-mounted cameras," SEISAN KENKYU, Vol.66, pp. 87-94, January 2015.

[12] S. Kasahara and J. Rekimoto, "Jackin head: An immersive human-human telepresence system," SIGGRAPH Asia 2015 Emerging Technologies, pp. 14:1-14:3, November 2015.

[13] A. Makadia, L. Sorgi, and K. Daniilidis, "Rotation estimation from spherical images," Proc. of the 17th Int. Conf. on Pattern Recognition 2004 (ICPR 2004), Vol.3, pp. 590-593, August 2004.

[14] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H. Y. Shum, "Full-frame video stabilization with motion inpainting," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.28, No.7, pp. 1150-1163, May 2006.

[15] L. Valgaerts, A. Bruhn, M. Mainberger, and J. Weickert, "Dense versus sparse approaches for estimating the fundamental matrix," Int. J of Computer Vision, Vol.96, pp. 212-234, January 2012.

[16] 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, pp. 261-273, March 1988.

[17] T. W. Hui and R. Chung, "Determining motion directly from normal flows upon the use of a spherical eye platform," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pp. 2267-2274, June 2013.

[18] Y. Yagi, W. Nishii, K. Yamazawa, and M. Yachida, "Stabilization for mobile robot by using omnidirectional optical flow," Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Vol.2, pp. 618-625, November 1996.

[19] S. Pathak, A. Moro, A. Yamashita, and H. Asama, "A decoupled virtual camera using spherical optical flow," Proc. of the IEEE Int. Conf. on Image Processing, September 2016.

[20] A. Torii, A. Imiya, and N. Ohnishi, "Two-and three-view geometry for spherical cameras," Proc. of the sixth workshop on omnidirectional vision, camera networks and non-classical cameras, pp. 81-88, October 2005.

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

[22] G. Farneback, "Two-frame motion estimation based on polynomial expansion," Lecture Notes in Computer Science, Vol.2749, pp. 363-370, June 2003.

[23] J. Fujiki, A. Torii, and S. Akaho, "Epipolar geometry via rectification of spherical images," Proc. of the Third Int. Conf. on Computer Vision/Computer Graphics Collaboration Techniques, pp. 461-471, March 2007.

[24] A. Pagani and D. Stricker, "Structure from motion using full spherical panoramic cameras," Proc. of the IEEE Int. Conf. on Computer Vision (Workshops), pp. 375-382, November 2011.

[25] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," J of the society for Industrial and Applied Mathematics, Vol.11, No.2, pp. 431-441, June 1963.

[26] H. Taira, Y. Inoue, A. Torii, and M. Okutomi, "Robust feature matching for distorted projection by spherical cameras," Information and Media Technologies, Vol.10, No.3, pp. 478-482, November 2015.

[27] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," Proc. of the British Machine Vision Conf., September 2013.

Supporting Online Materials:

[a] A few video results of our proposed stabilization can be found at: https://www.youtube.com/watch?v=d9tqw_ZlSC0 [Accessed May 30, 2017]
They were stabilized using the direct epipolar approach. The front and back view of the spherical image were unwarped to demonstrate the rotation-less virtual camera. It can be seen from the results that the stabilized videos are much easier to view as compared to the unstabilized ones.

Name:
Sarthak Pathak

Affiliation:
The University of Tokyo

Address:
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Brief Biographical History:

2014 Received Bachelor of Technology and Master of Technology degrees from Department of Engineering Design, Indian Institute of Technology, Madras (IITM)

2014- Doctoral Student, Department of Precision Engineering, The University of Tokyo

Main Works:

- omnidirectional vision, especially 3D reconstruction, SLAM, and stereo vision using them

Membership in Academic Societies:

- The Institute of Electrical and Electronic Engineers (IEEE) Robotics and Automation Society

Name:
Alessandro Moro

Affiliation:
The University of Tokyo
Ritecs Inc.

Address:
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Brief Biographical History:

2006 Graduated in Computer Science from the University of Udine

2011 Received Ph.D. from the University of Trieste (Italy)

2011- Visiting Research Fellow on Computer Vision, Chuo University

2012- Research Engineer, Ritecs Inc.

2012- Visiting Research Fellow on Computer Vision, The University of Tokyo

Main Works:

- 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.

Membership in Academic Societies:

- The Institute of Electrical and Electronic Engineers (IEEE)

Name:
Hiromitsu Fujii

Affiliation:
The University of Tokyo

Address:
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Brief Biographical History:

2005 Graduated in Precision Mechanical Engineering from The University of Tokyo
2007-2013 Image Processing Researcher and Software Engineer, Sony Corporation
2014-2016 Granted JSPS Research Fellowship
2016 Received Ph.D. in Precision Engineering from The University of Tokyo
2016-2017 Assistant Professor, Department of Precision Engineering, The University of Tokyo
2017- Lecturer, Department of Precision Engineering, The University of Tokyo

Main Works:

- signal processing for robotics systems, such as multi-sensor integration for unmanned tele-operation systems, and pattern recognition for automated diagnosis using robots

Membership in Academic Societies:

- The Institute of Electrical and Electronic Engineers (IEEE)
- ACM SIGGRAPH
- The Robotics Society of Japan (RSJ)
- The Japan Society of Mechanical Engineers (JSME)
- The Society of Instrument and Control Engineers (SICE)

Name:
Atsushi Yamashita

Affiliation:
The University of Tokyo

Address:
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Brief Biographical History:

1996/1998/2001 Received B.E., M.E., and Ph.D. degrees from Department of Precision Engineering, The University of Tokyo
1998-2001 Junior Research Associate, RIKEN (Institute of Physical and Chemical Research)
2001-2008 Assistant Professor, Shizuoka University
2006-2007 Visiting Associate, California Institute of Technology
2008-2011 Associate Professor, Shizuoka University
2011- Associate Professor, Department of Precision Engineering, The University of Tokyo

Main Works:

- robot vision, image processing, multiple mobile robot system, and motion planning

Membership in Academic Societies:

- Association for Computing Machinery (ACM)
- The Institute of Electrical and Electronics Engineers (IEEE)
- The Japan Society for Precision Engineering (JSPE)
- The Robotics Society of Japan (RSJ)
- The Institute of Electronics, Information and Communication Engineers (IEICE)
- The Japan Society of Mechanical Engineers (JSME)
- The Institute of Electrical Engineers of Japan (IEEJ)
- Information Processing Society of Japan (IPSJ)
- The Institute of Image Information and Television Engineers (ITE)
- The Society of Instrument and Control Engineers (SICE)

Name:
Hajime Asama

Affiliation:
The University of Tokyo

Address:
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Brief Biographical History:

1982/1984/1989 Received B.S., M.S., and Dr.Eng. degrees from The University of Tokyo
1986-2002 Research Associate, Research Scientist, and Senior Research Scientist, RIKEN (The Institute of Physical and Chemical Research, Japan)
2002- Professor, RACE (Research into Artifacts, Center for Engineering), The University of Tokyo
2009- Professor, School of Engineering, The University of Tokyo
2011- Project Leader, Disaster Response Robots and Their Operation System of Council on Competitiveness-Japan
2011- Chairman, Robotics Task Force for Anti-Disaster (ROBOTAD)
2012- Chairman, Task Force for Remote Control Technology of the Council for Decommissioning of TEPCO's Fukushima Daiichi NPS

Main Works:

- distributed autonomous robotic systems, smart spaces, service engineering, and mobiligence, and service robotics

Membership in Academic Societies:

- The Robotics Society of Japan (RSJ)
- The Japan Society of Mechanical Engineers (JSME)
- The Japan Society for Precision Engineering (JSPE)
- The Society of Instrument and Control Engineers (SICE)
- The Institute of Systems, Control and Information Engineers (ISCIE)
- The Japanese Society for Artificial Intelligence (JSAI)
- Information Processing Society of Japan (IPSJ)
- The Society of Chemical Engineers, Japan (SCEJ)
- The Institute of Electrical Engineers of Japan (IEEJ)
- The Atomic Energy Society of Japan (AESJ)
- Japan Robot Association (JARA)
- The Institute of Electrical and Electronics Engineers (IEEE)
- The New York Academy of Science (NYAS)
- International Federation of Automatic Control (IFAC)