Structure-from-motion in Spherical Video using the von Mises-Fisher Distribution

Hao Guan and William A. P. Smith, *Member, IEEE*

***Abstract*—In this paper we present a complete pipeline for computing structure-from-motion from sequences of spherical images. We revisit problems from multiview geometry in the context of spherical images. In particular, we propose methods suited to spherical camera geometry for the spherical-n-point problem (estimating camera pose for a spherical image) and calibrated spherical reconstruction (estimating the position of a 3D point from multiple spherical images). We introduce a new probabilistic interpretation of spherical structure-from-motion which uses the von Mises-Fisher distribution to model noise in spherical feature point positions. This model provides an alternate objective function that we use in bundle adjustment. We evaluate our methods quantitatively and qualitatively on both synthetic and real world data and show that our methods developed for spherical images outperform straightforward adaptations of methods developed for perspective images. As an application of our method, we use the structure-from-motion output to stabilise the viewing direction in fully spherical video.**

***Index Terms*—Structure-from-motion, view stabilisation, spherical image, 360 video.**

I. Introduction

SPHERICAL video (also known as 360, omnidirectional or surround video) [28] has recently gained tremendous popularity. Compared to traditional video, it benefits from a much greater sense of immersion [37] since a virtual viewing direction can be chosen and varied during playback. This rise in popularity has been fuelled by the availability of 360 cameras, support for 360 playback in web video services and the release of consumer virtual reality head mounted displays.

The benefits of spherical video and the sense of immersion are particularly felt in the case of first person (or egocentric) sequences. First person video captured with traditional narrow-field-of-view cameras suffered from rapid changes in viewing direction leading to a highly disorientating experience for the viewer. Spherical video offers a potential solution to this problem since all possible viewing directions are captured. However, as the pose of the spherical camera changes, so too does the virtual viewing direction. This is also disorienting for the viewer and can even lead to motion sickness as they lose control over the direction in which they are looking.

This motivates the need for algorithms that can robustly stabilise spherical video captured in the real world with large and rapid changes in camera pose. From a computer vision perspective, spherical video is in some ways attractive. The greatly increased field of view substantially increases the chance of observing features seen previously and the spherical camera projection model is particularly simple (it does not require intrinsic calibration). However, the adaptation of computer vision techniques such as feature extraction and structure-from-motion (SFM) to spherical images has received limited attention and the majority of previous work has heuristically adapted techniques developed for perspective images.

In this paper, we present a complete SFM pipeline that is specifically adapted to spherical image geometry and apply the method to the problem of spherical video stabilisation. There are a number of novel ingredients to our work. First, we present a probabilistic model of spherical image formation and use the von Mises-Fisher distribution to model spherical noise. Second, this leads us to a novel objective function for the spherical SFM problem. Third, in order to initialise minimisation of the non-convex objective function, we propose new methods for pose estimation and 3D world point reconstruction that are specifically adapted to the spherical case. Finally, we present a complete spherical SFM pipeline that combines these new methods with spherical feature match filtering for robustness.

We present quantitative experimental results on both synthetic and real world data. Our results show that our spherical pose estimation methods significantly outperform a straightforward adaptation of the classical direct linear transform (DLT) approach. Moreover, the complete pipeline that includes optimisation of the error term based on the von Mises-Fisher distribution outperforms squared angular error or squared Euclidean distance as used in previous work. Finally, we present qualitative stabilisation results on real world first person spherical video sequences. Using the camera pose estimates provided by SFM, we rotate the frames back onto a canonical view and are able to remove the effect of viewing direction changes. Our approach is sufficiently robust to work well on real world, highly unstable image sequences (stabilised videos are included as supplementary material). We will make a Matlab implementation of our SFM pipeline available upon acceptance of the paper.

II. Related Work

In this section we describe related work in the areas of spherical SFM, spherical features, video view stabilisation and the use of spherical probabilistic models in computer vision.

Spherical Structure-from-motion There have been a number of attempts to extend SFM to operate on spherical images. Typically, this is in the context of robotics, autonomous vehicles or mapping, where the wide field of view enables maps to be constructed more quickly and location estimated with less ambiguity. Some of the earliest work was by Chang et al. [7] who described the epipolar constraint for catadioptric cameras. They estimate the essential matrix using the standard approach

L. Guan and W. Smith are with the Department of Computer Science, University of York, UK. e-mail: {hg607,william.smith}@york.ac.uk.

of a linear initialisation followed by nonlinear optimisation. Their objective is identical to the perspective case and they only consider the relative pose problem (not 3D reconstruction or optimisation of multiple poses simultaneously). Torii et al. [35] made a similar theoretical contribution for the case of a central projection, spherical camera. They derived both two and three view geometric constraints.

Theoretical aspects of SFM in the context of spherical images have been considered by a number of researchers. Pagani et al. [30] presented straightforward modifications to perspective SFM to operate with the same spherical camera model that we use. The approach that they propose for pose estimation provides the baseline against which we compute. Pagani and Stricker [29] proposed several variations on the objective function used during spherical SFM. Their goal was to find error terms that could be efficiently evaluated yet agreed closely with minimising squared angular error. In contrast, the probabilistic model that we propose leads to an objective function based on maximisation of dot products. Krolla et al. [19] also consider different reprojection errors for spherical SFM, namely: angular (geodesic), Euclidean and tangential distance. The goal is to study uncertainty propagation in the context of optimisation (bundle adjustment) or spherical SFM problems. An interesting result is that geodesic distance results in corrupted uncertainty estimates. While useful from a practical perspective, we argue that these distance measures are not justified by an explicit noise or probabilistic model. Our contribution in this regard is to use the von Mises-Fisher distribution to model uncertainty in spherical image formation and use this derive a novel objective function.

Visual odometry and Simultaneous Location and Mapping (SLAM) have both been shown to benefit from omnidirectional cameras. Although they do not explicitly use a spherical camera model, Tardif et al. [34] present an omnidirectional SLAM using a fixed rig of perspective cameras. By combining efficient and robust landmark tracking and robust pose estimation, they are able to obtain very accurate trajectories for planar motion of a vehicle without a global bundle adjustment step. Schmeing et al. [32] take a different approach, stitching the multiple perspective images into a (partial) spherical image and proposing a bundle adjustment scheme using the same spherical camera model that we use. They directly adopt the energy term typically used in perspective bundle adjustment, namely sum of squared Euclidean reprojection distance. It is not clear that this is appropriate for spherical images. Gutierrez et al. [15] adapt a state-of-the-art, realtime SLAM system based on the Extended Kalman Filter to operate on omnidirectional video. Their method is based on linearisation of a catadioptric camera model and they perform tracking with omnidirectional patch descriptors that are rotation and scale invariant. Murillo et al. [27] apply the same approach to data that is similar to that in our stabilisation application, namely first person video captured by a wearable omnidirectional camera. Torii et al. [36] build 3D city models from Google Street View imagery. Their focus is on building a robust, scalable system applicable to large numbers of high resolution images. Relative pose is computed using standard modifications of perspective methods but they introduce a robust technique for estimating the scale of translations. Bundle adjustment minimises an angular error, as in [20].

There have been a number of attempts to build hybrid SFM pipelines that incorporate or unify multiple camera geometries. Bastanlar et al. [4] use a central catadioptric model that can also model perspective cameras. This allows them to perform SFM on pairs of catadioptric and perspective images and they propose a preprocessing step to enable standard feature descriptors to be matched between the different image modalities. To enable a linear expression of the epipolar geometry, they use lifted coordinates for omnidirectional image points. Recently, Gava and Stricker [10] proposed a unified SFM framework that allows uniform treatment of central projection cameras using a sphere as the underlying model. This allows perspective, spherical and hybrid image datasets to be analysed. Lhuillier [20] proposed minimisation of angular error during bundle adjustment as a means to unify multiple camera geometries.

In general, previous work has considered various omnidirectional camera models for SFM and heuristically adapted steps such as relative pose estimation, the n-point problem, triangulation and bundle adjustment from methods used for perspective images. In contrast, in this paper we begin with a probabilistic model that is specific to spherical images and use this to derive novel objective functions for spherical SFM. We also propose modifications to standard methods for pose estimation and triangulation that are appropriate for spherical images where features are observed in all directions. Our evaluation shows that these modifications lead to measurable quantitative improvement in accuracy.

Spherical Feature Descriptors A recent development has been the creation of interest points and local feature descriptors that account for the geometry of spherical images. Simply using 2D local features on 2D parameterisations of spherical images is unreliable due to the directionally-dependent distortion present. Various authors have previously overcome this problem via pre-processing or simple hacks to improve performance of planar descriptors on spherical images, e.g. [30] generate multiple virtual perspective views and then extract planar features. However, recent work has taken a more principled approach. Cruz-Mota et al. [8] extend scale space theory to the sphere and use this develop a spherical extension of SIFT. We refer to this as SSIFT and use it in our proposed pipeline. More recently, Guan and Smith [14] extended the accelerated segment test for interest point detection to a geodesic grid on the sphere. In the same direction, Zhao et al. [39] built on the popularity of binary descriptors and developed a spherical extension of the ORB descriptor, again on a geodesic spherical grid.

View Stabilisation Our motivating application is that of stabilising first person, spherical video sequences. View stabilisation for video is a well studied problem for the case where images are captured by a perspective camera. Approaches can be roughly divided into those that estimate and apply a single homography to each frame [11] and those that construct an explicit 3D model and use this in the stabilisation process

[21]. The former is easier to estimate and less computationally expensive but can only correctly stabilise scenes that are largely planar. The latter can potentially stabilise any motion through complex scenes but requires estimation of a 3D model (usually via SFM), a process that is computationally expensive and potentially fragile.

With an estimate of the 3D scene to hand, Liu et al. [21] compute a new stabilised path through the scene and use a grid-mesh to warp the input images in a content-preserving manner. A popular recent alternative is to use 2D feature trajectories to guide the warp without explicitly estimating 3D scene information. Liu et al. [22] pose the problem of smoothing feature trajectories as one of low rank matrix factorisation. State-of-the-art approaches use feature trajectories to compute bundles of homographies and apply mesh-based warping [23]. This idea was extended to include user-guided constraints on the stabilisation result [2].

In these methods, since they use perspective views, the range of possible viewing directions is limited and the quality of the output depends on whether desirable viewing directions were sampled in the input. The only previous work that we are aware of that is applicable to spherical video sequences was due to Kamali et al. [18]. They use spherical SFM to compute sparse 3D information for pose estimation. The stabilisation process relies on a mesh-based warp by reprojection of the 3D points. Their SFM pipeline is based on simple adaptations of a perspective image pipeline and does not consider the differences inherent in spherical image geometry.

Spherical Distributions Our probabilistic model for spherical SFM uses the von Mises-Fisher (vMF) distribution [9] which has been widely used in computer vision and machine learning. For example, Banerjee et al. [3] use the Expectation Maximisation algorithm for clustering spherical data according to the vMF. In particular, the vMF has been used in applications related to multiview vision and, more specifically, omnidirectional vision. Pons-Moll et al. [31] present a system for multiview human motion capture in which sensor noise is modelled with the vMF. Ćesić et al. [6] use the vMF for object tracking in a multi-camera system. Markovic et al. [25] extend the approach to spherical images. Bazin et al. [5] use diffusion of particles on the sphere for tracking in catadioptric images using a particle tracking framework. We are not aware of the vMF having been used previously for spherical SFM.

III. Preliminaries

The projection of a 3D world point, $\mathbf{w} = [u\ v\ w]^T$, to a point in a spherical image, $\mathbf{x} = [x\ y\ z]^T$ (with $\|\mathbf{x}\| = 1$), for a spherical camera whose coordinate system differs from world coordinates by a rotation $\boldsymbol{\Omega} \in SO(3)$ and translation $\boldsymbol{\tau} \in \mathbb{R}^3$ is given by the spherical camera model:

$$\mathbf{x} = \textbf{spherical}[\mathbf{w}, \boldsymbol{\Omega}, \boldsymbol{\tau}] = \frac{\boldsymbol{\Omega}\mathbf{w} + \boldsymbol{\tau}}{\|\boldsymbol{\Omega}\mathbf{w} + \boldsymbol{\tau}\|}. \tag{1}$$

Note that this model is quite general (it is applicable to catadioptric and dioptric cameras or spherical panoramas obtained via stitching [26], [33]) and abstracts away from how the input images were acquired.

Unlike a pinhole (perspective) camera, a spherical camera has no intrinsic parameters. Yet there is a close relationship between a pinhole and spherical camera. In a pinhole camera, projection involves rescaling rays such that they lie on the image plane. In a spherical camera, rays are normalised to lie on the unit sphere. Hence, pinhole image points are 2D, usually represented using homogeneous coordinates. Spherical image points are unit vectors in $\mathbb{R}^3$ (i.e. points on the S^2 sphere).

A. Noise Model

The estimated position of a feature in a spherical image is noisy for reasons including sensor noise, sampling issues (exacerbated by stitching or warping to obtain the full spherical image) and inaccuracies of the feature detector. The vMF distribution is a parametric distribution for directional data and has properties analogous to those of the multi-variate Gaussian distribution for data in $\mathbb{R}^n$. Hence, where the assumption of additive, normally distributed noise with spherical covariance is assumed for perspective projection of 3D points to a 2D image plane, so the vMF distribution is appropriate for 3D points projected to the unit 2-sphere.

For the 2-sphere, the PDF of the vMF for the random unit vector $\mathbf{x}$, $\|\mathbf{x}\| = 1$ is given by:

$$\text{vMF}_{\mathbf{x}}[\boldsymbol{\mu}, \kappa] = C(\kappa)\text{exp}(\kappa\boldsymbol{\mu}^T\mathbf{x}), \tag{2}$$

where the normalisation constant is given by:

$$C(\kappa) = \frac{\kappa}{4\pi\sinh\kappa}, \tag{3}$$

$\boldsymbol{\mu} \in \mathbb{R}^3$ is the mean direction ($\|\boldsymbol{\mu}\| = 1$) and $\kappa \in \mathbb{R}$ is the concentration parameter.

Under the assumption of vMF-distributed noise, we can write a probabilistic spherical camera model as:

$$\Pr(\mathbf{x}|\mathbf{w}, \boldsymbol{\Omega}, \boldsymbol{\tau}) = \text{vMF}_{\mathbf{x}}[\textbf{spherical}[\mathbf{w}, \boldsymbol{\Omega}, \boldsymbol{\tau}], \kappa]. \tag{4}$$

Hence, the expected position of a spherical image point is the projection of the corresponding 3D point via the spherical camera model (1) but the observation is subject to vMF noise about this position.

IV. Spherical Structure From Motion

The goal of spherical structure from motion is to choose the most likely 3D positions of the I observed points and the pose of the J cameras that observed those points. The maximum likelihood solution with vMF noise is given by:

$$\hat{\theta} = \arg\max_{\theta}\left[\sum_{i=1}^{I}\sum_{j=1}^{J}\log[\Pr(\mathbf{x}_{ij}|\mathbf{w}_i, \boldsymbol{\Omega}_j, \boldsymbol{\tau}_j)]\right] \tag{5}$$

$$= \arg\max_{\theta}\sum_{i=1}^{I}\sum_{j=1}^{J}\log[\text{vMF}_{\mathbf{x}_{ij}}[\textbf{spherical}[\mathbf{w}_i, \boldsymbol{\Omega}_j, \boldsymbol{\tau}_j], \kappa]] \tag{6}$$

$$= \arg\max_{\theta}\sum_{i=1}^{I}\sum_{j=1}^{J}\left(\frac{\boldsymbol{\Omega}_j\mathbf{w}_i + \boldsymbol{\tau}_j}{\|\boldsymbol{\Omega}_j\mathbf{w}_i + \boldsymbol{\tau}_j\|}\right)^T\mathbf{x}_{ij}, \tag{7}$$

where θ contains the unknown world points $\{\mathbf{w}_i\}_{i=1}^{I}$ and the extrinsic parameters for each camera $\{\mathbf{\Omega}_j, \boldsymbol{\tau}_j\}_{j=1}^{J}$. In practice, not all cameras observe all features so a feature matrix is used to keep track of this and the summation is only over 3D points with corresponding observations.

Hence, the solution maximises the dot product between the unit vector in the direction of the estimated 3D point position and the observed unit vector on the sphere. Contrast this with the error term used in previous attempts at spherical SFM (e.g. [18], [20], [29]) which minimises total squared angular error:

$$\theta_{\text{ang}} = \arg\min_{\theta} \sum_{i=1}^{I} \sum_{j=1}^{J} \arccos^2 \left[\left(\frac{\mathbf{\Omega}_j \mathbf{w}_i + \boldsymbol{\tau}_j}{\|\mathbf{\Omega}_j \mathbf{w}_i + \boldsymbol{\tau}_j\|} \right)^T \mathbf{x}_{ij} \right]. \tag{8}$$

While this objective is intuitive, it is not justified by an explicit spherical noise model. Moreover, it requires an additional inverse trigonometric function and exponentiation for each term compared to the vMF-derived objective function.

A. Pipeline

We begin by providing a high level overview of our spherical SFM pipeline. In the following subsections we then present in detail the theory associated with each step of the pipeline and our novel formulations of the spherical-n-point problem and calibrated spherical reconstruction. Finally, we provide pseudocode to describe the algorithm in more detail.

The general pipeline is as follows:

1) Select a pair of images, robustly estimate the spherical essential matrix and decompose it to find the rotation and translation. Estimate position of observed world points using calibrated spherical reconstruction.
2) Align new image to 3D world points visible in that image by solving the spherical-n-point problem.
3) Reconstruct 3D world points for which the new view provides a second observation by solving the calibrated spherical reconstruction problem.
4) Perform bundle adjustment over all estimated values.
5) Repeat steps 2-4 until all images included.

To resolve the unknown scale ambiguity, the length of the first translation is fixed to 1 and this is enforced during nonlinear optimisation. During nonlinear optimisations involving pose estimates, we represent rotation matrices in axis-angle space.

B. Two view spherical geometry

Two view epipolar geometry for spherical images closely follows that of the planar perspective formulation. Torii et al. [35] showed that if $\mathbf{x}$ and $\mathbf{x}'$ are two unit vectors representing corresponding points in two spherical images, then the essential matrix $\mathbf{E}$ satisfies: $\mathbf{x}^T\mathbf{E}\mathbf{x}' = 0$. Note that since spherical cameras do not have intrinsic parameters, there is no distinction between the essential and fundamental matrices. Also, in contrast to the perspective case, rather than being 2D image points represented as 3D homogeneous coordinates, $\mathbf{x}$ and $\mathbf{x}'$ are unit vectors in $\mathbb{R}^3$.

The essential matrix can be decomposed [16] as: $\mathbf{E} = [\boldsymbol{\tau}]_\times \mathbf{\Omega}$, where $\mathbf{\Omega}$ and $\boldsymbol{\tau}$ are the rotation and translation which relate the camera coordinate systems between the two views and $[.]_\times$ is the cross product matrix:

$$[\mathbf{x}]_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}. \tag{9}$$

To perform this decomposition [16], we take the singular value decomposition (SVD) $\mathbf{E} = \mathbf{U}\mathbf{L}\mathbf{V}^T$. The translation is given by $\boldsymbol{\tau} = \pm\mathbf{u}_3$ and the rotation matrix by $\mathbf{\Omega} = \mathbf{U}\mathbf{W}^{-1}\mathbf{V}^T$ or $\mathbf{\Omega} = \mathbf{U}\mathbf{W}\mathbf{V}^T$ where

$$\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

Hence, there are four possible combinations of translation vector and rotation matrix. In the perspective case, this fourfold ambiguity is resolved by choosing the solution which places points in front of the camera. For the spherical case, we propose a slightly different procedure. It is useful to rewrite the spherical camera equation for the first two views in terms of scale parameters λ and λ':

$$\lambda\mathbf{x} = \mathbf{w} \quad \text{and} \quad \lambda'\mathbf{x}' = \mathbf{\Omega}\mathbf{w} + \boldsymbol{\tau}, \tag{11}$$

where $\lambda, \lambda' > 0$. Spherical point $\mathbf{x}$ is observed by the first camera whose coordinate system coincides with world coordinates. Substituting and rearranging provides three linear equations in terms of the two scale parameters:

$$\lambda'\mathbf{x}' - \lambda\mathbf{\Omega}\mathbf{x} - \boldsymbol{\tau} = \mathbf{0}. \tag{12}$$

We use this equation to select from the four possible combinations of rotation matrix and translation vector extracted from $\mathbf{E}$. Since we expect both scale parameters to be positive, for each point pair we substitute each of the four $(\mathbf{\Omega}, \boldsymbol{\tau})$ and solve for λ and λ'. If both are positive, we cast a vote for the solution $(\mathbf{\Omega}, \boldsymbol{\tau})$. The solution with the most votes is chosen. Note that the substitution to obtain (12) eliminates $\mathbf{w}$ and allows us to test the positivity of λ and λ' without first solving for $\mathbf{w}$.

In practice, for robustness we use Random Sample Consensus (RANSAC) to estimate the essential matrix from point correspondences between the first pair of images. We then decompose as above to obtain the rotation and translation between these two images.

C. The spherical-n-point problem

For images 3 and onwards, we estimate their pose relative to the current estimate of the 3D scene. Estimating camera pose (i.e. extrinsic parameters) relative to a known 3D scene is a well-studied problem for perspective images and is known as the perspective-n-point (PnP) problem. We propose here a variant of this problem adapted specifically to spherical images and refer to it as the spherical-n-point (SnP) problem.

Given I 3D world points and their corresponding spherical projections, the maximum likelihood solution for the extrinsic parameters of the new camera is given by:

$$\hat{\mathbf{\Omega}}, \hat{\boldsymbol{\tau}} = \underset{\mathbf{\Omega}\in\text{SO}(3), \boldsymbol{\tau}\in\mathbb{R}^3}{\arg\max} \sum_{i=1}^{I} \left(\frac{\mathbf{\Omega}\mathbf{w}_i + \boldsymbol{\tau}}{\|\mathbf{\Omega}\mathbf{w}_i + \boldsymbol{\tau}\|} \right)^T \mathbf{x}_i. \tag{13}$$

This is an optimisation problem that is constrained (since $\mathbf{\Omega}$ must be a rotation matrix) and non-convex. Hence, the globally optimal solution cannot be found in closed form and optimisation may only provide a local minima. The goal therefore is to develop a robust and efficient means to compute a good estimate of the extrinsic parameters which can be used to initialise a non-linear optimisation of (13).

The classical approach for computing initial estimates for the rotation and translation (in both perspective and spherical SFM) is the DLT method (originally so-called for the perspective case by Abdel-Aziz and Karara [1]). This is a linearisation of the non-convex objective based on a collinearity condition between a 3D point and its projection. From the same starting point, we derive a modification of the standard DLT that is specifically adapted to spherical image geometry.

Following the DLT method, we can express the collinearity condition for each visible scene point as follows:

$$\lambda \mathbf{x}_i = \mathbf{\Omega} \mathbf{w}_i + \boldsymbol{\tau}. \tag{14}$$

This is a linear similarity relation whose solution is also a solution to the following linear equation:

$$\mathbf{x}_i \times (\mathbf{\Omega} \mathbf{w}_i + \boldsymbol{\tau}) = \mathbf{0}. \tag{15}$$

Intuitively, (15) says that the vector to the 3D point in the camera coordinate system should be parallel to the unit vector to the corresponding spherical point (i.e. their cross product is the zero vector). Each 3D point contributes three equations to a system of linear equations (although note that since the cross product matrix has rank 2, only two of the equations are linearly independent). The complete system of equations can be expressed as a homogeneous system:

$$\mathbf{A}\mathbf{b} = \mathbf{0}, \tag{16}$$

where

$$\mathbf{A} = \begin{bmatrix} [\mathbf{x}_1]_\times \begin{bmatrix} u_1 & v_1 & w_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & u_1 & v_1 & w_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & u_1 & v_1 & w_1 & 1 \end{bmatrix} \\ \vdots \\ [\mathbf{x}_I]_\times \begin{bmatrix} u_I & v_I & w_I & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & u_I & v_I & w_I & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & u_I & v_I & w_I & 1 \end{bmatrix} \end{bmatrix},$$

and the vector

$$\mathbf{b} = [\omega_{11} \; \omega_{12} \; \omega_{13} \; \tau_x \; \omega_{21} \; \omega_{22} \; \omega_{23} \; \tau_y \; \omega_{31} \; \omega_{32} \; \omega_{33} \; \tau_z]^T$$

contains a vectorised version of the rotation matrix and translation vector (ω_{ij} refers to the (i,j)th element of the rotation matrix $\mathbf{\Omega}$ and $\boldsymbol{\tau} = [\tau_x \; \tau_y \; \tau_z]^T$). Since this is a homogeneous system, we observe that: 1. there is always a trivial solution $\mathbf{b} = \mathbf{0}$, 2. if $\mathbf{b} \neq \mathbf{0}$ is a solution then $k\mathbf{b}$ is also a solution.

Due to noise, we do not expect an exact solution to this problem. Hence, we may instead minimise a least squares criterion: $\|\mathbf{A}\mathbf{b}\|^2$. In order to resolve the arbitrary scaling and to avoid the trivial solution, the standard approach is to solve a minimum direction problem of the form: minimise $\|\mathbf{A}\mathbf{b}\|^2$ subject to $\|\mathbf{b}\| = 1$. This minimisation problem is straightforward to solve using an SVD of $\mathbf{A}$. This imposes no constraint that the elements of $\mathbf{\Omega}$ should form a valid rotation matrix. Hence, the elements of $\mathbf{b}$ corresponding to the rotation matrix are transformed to the closest orthogonal matrix by solving an orthogonal Procrustes problem. The scale of the translation is given by the mean scale between the estimated rotation matrix and the raw estimate taken from $\mathbf{b}$.

Finally, there is a sign ambiguity that must be resolved: negating the rotation and translation still satisfies the collinearity condition in (14). This ambiguity arises because the condition is minimised both by a vector pointing in the same direction as a spherical point, but also its negative. However, only one of the two possible solutions will yield a valid rotation matrix. We test whether the determinant of the rotation matrix is positive and, if not, negate both the translation vector and rotation matrix. This is the baseline method (used previously [30]) against which we compare in our experimental evaluation.

The drawback to this approach is that there is no guarantee of positive depth for all points or even a large majority. In practice, for spherical images (where features may be observed in all directions) we observe that the classical DLT often aligns a 3D point and the corresponding spherical point in opposite directions. For this reason, we propose instead to integrate the depth test into the optimisation. To do so, we impose positive depth:

$$\mathbf{x}_i \cdot (\mathbf{\Omega} \mathbf{w}_i + \boldsymbol{\tau}) \geq 0. \tag{17}$$

as either a soft or hard constraint on the solution of (16). Importantly, both of our formulations remain convex optimisation problems and we show that they outperform the classical DLT approach.

1) Hard constraint: We can express the dot product constraints in matrix form as $\mathbf{C}\mathbf{b} \geq \mathbf{0}$ where

$$\mathbf{C} = \begin{bmatrix} u_1x_1 & v_1x_1 & w_1x_1 & x_1 & u_1y_1 & v_1y_1 & w_1y_1 & y_1 & u_1z_1 & v_1z_1 & w_1z_1 & z_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_Ix_I & v_Ix_I & w_Ix_I & x_I & u_Iy_I & v_Iy_I & w_Iy_I & y_I & u_Iz_I & v_Iz_I & w_Iz_I & z_I \end{bmatrix}.$$

Hence, the constrained minimisation problem is

$$\min_{\mathbf{b}} \|\mathbf{A}\mathbf{b}\|^2 \quad \text{s.t.} \; -\mathbf{C}\mathbf{b} \leq \mathbf{0}.$$

This is a standard inequality constrained homogeneous least squares problem. Unfortunately, it still has a trivial solution $\mathbf{b} = \mathbf{0}$. Imposing the quadratic equality constraint $\|\mathbf{b}\| = 1$ as for the DLT method leads to a quadratically-constrained quadratic programming (QCQP) problem. This is non-convex like the original optimisation problem (13) for which we sought a convex initialisation. Instead, we impose a simple linear equality constraint on one element of $\mathbf{b}$. Note that all elements of $\boldsymbol{\tau}$ (stored in b_4, b_8 and b_{12}) could be zero and elements of $\mathbf{\Omega}$ (stored in $b_{1..3}$, $b_{5..7}$ and $b_{9..11}$) could be zero or negative. So forcing one element to unity may lead to a very poor solution.

For this reason, we solve the problem six times with different linear equality constraints $b_1 = \pm 1$, $b_2 = \pm 1$ and $b_3 = \pm 1$ (since a row of a rotation matrix must contain at least one non-zero entry). Which ever solution gives the lowest residual error (once the appropriate elements of $\mathbf{b}$ have been transformed to the closest rotation matrix) is taken as the solution. Explicitly, we solve the following constrained linear least squares problem:

$$\min_{\mathbf{b}} \|\mathbf{Ab}\|^2 \ \text{ s.t.} -\mathbf{Cb} \leq \mathbf{0} \wedge (b_1 = \pm 1 \vee b_2 = \pm 1 \vee b_3 = \pm 1), \tag{18}$$

and recover $\boldsymbol{\Omega}$ and $\boldsymbol{\tau}$ from $\mathbf{b}$ as for the standard DLT method.

2) Soft constraint: Imposing positive depth as a hard constraint may force the estimated camera pose to be highly inconsistent with other measurements. For example, this is particularly the case when the data contains correspondence errors between 3D world points and spherical image points. For this reason, we propose a variant in which negative depths are penalised as a soft constraint. To do so, we penalise the square of any negative dot products:

$$\min_{\mathbf{b}} \|\mathbf{Ab}\|^2 + \gamma \| \max\{\mathbf{0}, -\mathbf{Cb}\} \|^2,$$
$$\text{s.t. } b_1 = \pm 1 \vee b_2 = \pm 1 \vee b_3 = \pm 1,$$

where the max operation is applied component-wise and γ weights the penalty term (we use $\gamma = 1$ in our experiments). The penalty term amounts to the sum of the square of the positive values of $-\mathbf{Cb}$. Importantly, this is still convex and we solve it using CVX, a package for specifying and solving convex programs [12], [13].

3) Weights: Finally, we consider a weighted variant of our method that allows the convex optimisation to be related back to our original objective function.

Although it is not immediately apparent, (15) is implicitly applying weights to each of the sets of linear equations. Expanding the cross product makes this clearer:

$$\min_{\boldsymbol{\Omega},\boldsymbol{\tau}} \sum_i \|\mathbf{x}_i \times (\boldsymbol{\Omega}\mathbf{w}_i + \boldsymbol{\tau})\|^2 \tag{19}$$
$$= \min_{\boldsymbol{\Omega},\boldsymbol{\tau}} \sum_i \| \ \sin(\phi_i)\|\mathbf{x}_i\|\|\boldsymbol{\Omega}\mathbf{w}_i + \boldsymbol{\tau}\|\mathbf{m}_i \ \|^2,$$

where ϕ_i is the angle between $\mathbf{x}_i$ and $\boldsymbol{\Omega}\mathbf{w}_i + \boldsymbol{\tau}$ and $\mathbf{m}_i$ is a unit vector orthogonal to $\mathbf{x}_i$ and $\boldsymbol{\Omega}\mathbf{w}_i + \boldsymbol{\tau}$. Since $\|\mathbf{x}_i\| = 1$ and the direction of $\mathbf{m}_i$ does not affect the magnitude of the expression, this simplifies to:

$$\min_{\boldsymbol{\Omega},\boldsymbol{\tau}} \sum_i \|\boldsymbol{\Omega}\mathbf{w}_i + \boldsymbol{\tau}\|^2 \sin^2(\phi_i). \tag{20}$$

It now becomes clear that by minimising the cross product, we actually minimise an angular error (the square of the sine of the angle) weighted by $\|\boldsymbol{\Omega}\mathbf{w}_i + \boldsymbol{\tau}\|^2$, i.e. the square of the Euclidean distance from the camera to the point. What this means is that points that are further from the camera are weighted more heavily in the optimisation. This is unlikely to be desirable since the accuracy of feature detection and matching is likely to degrade for points that are further away.

Let us now introduce weights into the minimisation:

$$\min_{\boldsymbol{\Omega},\boldsymbol{\tau}} \sum_i k_i \|\mathbf{x}_i \times (\boldsymbol{\Omega}\mathbf{w}_i + \boldsymbol{\tau})\|^2, \tag{21}$$

and define the weights as: $k_i = \|\boldsymbol{\Omega}\mathbf{w}_i + \boldsymbol{\tau}\|^{-2}$. Following the same derivation as above and writing in terms of cosine by the Pythagorean identity, this is equivalent to:

$$\max_{\boldsymbol{\Omega},\boldsymbol{\tau}} \sum_i \cos^2(\phi_i) = \max_{\boldsymbol{\Omega},\boldsymbol{\tau}} \sum_i \left[\left(\frac{\boldsymbol{\Omega}\mathbf{w}_i + \boldsymbol{\tau}}{\|\boldsymbol{\Omega}\mathbf{w}_i + \boldsymbol{\tau}\|} \right)^T \mathbf{x}_i \right]^2 \tag{22}$$

We now see a close relationship to the probabilistic formulation in (13). Namely, the weighted cross product objective differs from the probabilistic objective only in the fact that it squares the dot product terms. In practice however, we cannot compute the desired weights in since this requires the rotation and translation of the camera to already be known. Hence, we propose an iterative reweighting approach. First, we use the unweighted version to compute an initial rotation and translation estimate. We use this to compute weights for each point and then re-estimate rotation and translation using the weighted version. This process can be iterated to convergence and used with both the soft and hard constraints.

D. Calibrated Spherical Reconstruction

With estimates of the poses of J cameras to hand, the 3D position of a point $\mathbf{w}$ observed by those cameras can be computed by maximising likelihood with respect to $\mathbf{w}$:

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \sum_{j=1}^{J} \left(\frac{\boldsymbol{\Omega}_j \mathbf{w} + \boldsymbol{\tau}_j}{\|\boldsymbol{\Omega}_j \mathbf{w} + \boldsymbol{\tau}_j\|} \right)^T \mathbf{x}_j. \tag{23}$$

Again, following the DLT approach we write the collinearity criterion as a cross product: $[\mathbf{x}_j]_\times (\boldsymbol{\Omega}_j \mathbf{w} + \boldsymbol{\tau}_j) = \mathbf{0}$. and rewrite this as a system of J linear equations in terms of the unknown 3D point position $\mathbf{w}$: $\mathbf{Bw} = \mathbf{d}$, where

$$\mathbf{B} = \begin{bmatrix} \omega^1_{31} y_1 - \omega^1_{21} z_1 & \omega^1_{32} y_1 - \omega^1_{22} z_1 & \omega^1_{33} y_1 - \omega^1_{23} z_1 \\ \omega^1_{11} z_1 - \omega^1_{31} x_1 & \omega^1_{12} z_1 - \omega^1_{32} x_1 & \omega^1_{13} z_1 - \omega^1_{33} x_1 \\ \omega^1_{21} x_1 - \omega^1_{11} y_1 & \omega^1_{22} x_1 - \omega^1_{12} y_1 & \omega^1_{23} x_1 - \omega^1_{13} y_1 \\ \vdots & \vdots & \vdots \\ \omega^J_{31} y_J - \omega^J_{21} z_J & \omega^J_{32} y_J - \omega^J_{22} z_J & \omega^J_{33} y_J - \omega^J_{23} z_J \\ \omega^J_{11} z_J - \omega^J_{31} x_J & \omega^J_{12} z_J - \omega^J_{32} x_J & \omega^J_{13} z_J - \omega^J_{33} x_J \\ \omega^J_{21} x_J - \omega^J_{11} y_J & \omega^J_{22} x_J - \omega^J_{12} y_J & \omega^J_{23} x_J - \omega^J_{13} y_J \end{bmatrix}, \tag{24}$$

and

$$\mathbf{d} = \begin{bmatrix} \tau^1_z y_1 - \tau^1_y z_1 \\ \tau^1_x z_1 - \tau^1_z x_1 \\ \tau^1_y x_1 - \tau^1_x y_1 \\ \vdots \\ \tau^J_z y_J - \tau^J_y z_J \\ \tau^J_x z_J - \tau^J_z x_J \\ \tau^J_y x_J - \tau^J_x y_J \end{bmatrix}. \tag{25}$$

The superscripts on ω and τ indicate with which camera the extrinsic parameters are associated.

We again solve in a least squares sense by minimising $\|\mathbf{Bw} - \mathbf{d}\|^2$. Unlike the SnP problem, this linear system is not homogeneous and hence constraints to avoid a trivial solution are not required. However, the same problem arises that the collinearity condition is satisfied by placing a 3D point in the opposite direction of a spherical point and so we use the same hard or soft constraints as above.

To do so, we rewrite the dot product constraint in (17) in terms of $\mathbf{w}$ and stack the J equations in a system of linear equations, yielding the following inequality constraint: $-\mathbf{Fw} \leq \mathbf{g}$, where

$$\mathbf{F} = \begin{bmatrix} \omega^1_{11} x_1 + \omega^1_{21} y_1 + \omega^1_{31} z_1 & \omega^1_{12} x_1 + \omega^1_{22} y_1 + \omega^1_{32} z_1 & \omega^1_{13} x_1 + \omega^1_{23} y_1 + \omega^1_{33} z_1 \\ \vdots & \vdots & \vdots \\ \omega^J_{11} x_1 + \omega^J_{21} y_1 + \omega^J_{31} z_1 & \omega^J_{12} x_1 + \omega^J_{22} y_1 + \omega^J_{32} z_1 & \omega^J_{13} x_1 + \omega^J_{23} y_1 + \omega^J_{33} z_1 \end{bmatrix},$$

and

$$\mathbf{g} = \begin{bmatrix} \tau_x^1 x_1 + \tau_y^1 y_1 + \tau_z^1 z_1 \\ \vdots \\ \tau_x^J x_J + \tau_y^J y_J + \tau_z^J z_J \end{bmatrix}.$$

This can be enforced as either a hard or soft constraint in exactly the same way as for the SnP methods described above.

Algorithm 1 Spherical Structure-from-motion

```
 1: Extract SSIFT [8] features for images 1 and 2
 2: Find matches $M_{1,2} = \{(a,b) | \mathbf{x}_{a1}$ matches $\mathbf{x}_{b2}\}$.[1]
    // (Optional) Remove possible non-scene points:
 3: for $(a,b)$ in $M_{1,2}$ do
 4:     if $\mathbf{x}_{a1} \cdot \mathbf{x}_{b2} > t_2$ then
 5:         $M_{1,2} := M_{1,2} \setminus \{(a,b)\}$
 6:     end if
 7: end for
 8: Compute pose transformation from view 1 to view 2 (refer
    Section IV-B).
 9: Estimate 3D world points $\{\mathbf{w}_i\}_{i=1}^{|M_{1,2}|}$ using calibrated
    spherical reconstruction (refer Section IV-D).
    // Remove noisy points:
10: for $j := 1$ to 2 do
11:     for $i := 1$ to $|M_{1,2}|$ do
12:         if $(\textbf{spherical}[\mathbf{w}_i, \mathbf{\Omega}_j, \boldsymbol{\tau}_j] \cdot \mathbf{x}_{ij}) < t_3$ then
13:             Remove $\mathbf{x}_{i1}$, $\mathbf{x}_{i2}$ and $\mathbf{w}_i$ from the reconstruction.
14:         end if
15:     end for
16: end for
17: Bundle adjustment of 3D world points and camera pose
    via nonlinear optimisation of (7).
18: for $j := 3$ to $J$ do
19:     Extract SSIFT [8] features for image $j$
20:     Find matches $M_{j,j-1} = \{(a,b) | \mathbf{x}_{a,j}$ matches $\mathbf{x}_{b,j-1}\}$.[1]
        // (Optional) Remove possible non-scene points:
21:     for $(a,b)$ in $M_{j,j-1}$ do
22:         if $\mathbf{x}_{a,j} \cdot \mathbf{x}_{b,j-1} > t_2$ then
23:             $M_{j,j-1} := M_{j,j-1} \setminus \{(a,b)\}$
24:         end if
25:     end for
        // Set of previously observed features:
26:     $M_{\text{prev}} := \{a | (a,b) \in M_{j,j-1} \wedge (b,c) \in M_{j-1,j-2}\}$
        // Set of newly observed features:
27:     $M_{\text{new}} := \{a | (a,b) \in M_{j,j-1} \wedge a \notin M_{\text{prev}}\}$
28:     Compute initial estimate of $\mathbf{\Omega}_j, \boldsymbol{\tau}_j$ by solving SnP on
        the set $\{(\mathbf{x}_{aj}, \mathbf{w}_a) | a \in M_{\text{prev}}\}$ (refer Section IV-C).
29:     Refine estimate of $\mathbf{\Omega}_j, \boldsymbol{\tau}_j$ by nonlinear optimisation of
        (13).
30:     Estimate new 3D world points $\{\mathbf{w}_i\}_{i \in M_{\text{new}}}$ using cali-
        brated spherical reconstruction (refer Section IV-D).
        // Remove noisy points:
31:     for $i$ in $M_{\text{new}}$ do
32:         if $(\textbf{spherical}[\mathbf{w}_i, \mathbf{\Omega}_j, \boldsymbol{\tau}_j] \cdot \mathbf{x}_{ij}) < t_3$ then
33:             Remove $\mathbf{x}_{ij}$ and $\mathbf{w}_i$ from the reconstruction.
34:         end if
35:     end for
36:     Bundle adjustment of 3D world points and camera poses
        2 to $j$ via nonlinear optimisation of (7).
37: end for
```

E. Implementation

We show our complete SFM pipeline in Algorithm 1. For efficiency, we only compute feature matches between adjacent images in a sequence. This is adequate for our goal of view stabilisation. For denser scene reconstruction, it would be necessary to also test for feature matches between a new image and earlier images in the sequence. Also, we impose no smoothness constraint on the camera poses in the sequence. Although this would likely improve results, it would also obscure the accuracy of pose estimates obtained solely from our proposed SnP and SFM pipeline. Finally, initialisation from the first two images may not always be a good choice when there is little motion between the first two frames.

Our algorithm relies on the selection of three parameters. t_1 is the threshold on feature distance ratios and determines the quality of match required for a feature to be included in the reconstruction (we use $t_1 = 0.75$ in our experiments). t_3 is the threshold on re-projection error and is used to filter outliers (we use $t_3 = \cos 10° = 0.985$).

The parameter t_2 relates to the removal of "non-scene" points. In the case of egocentric image sequences (i.e. 'first' or 'third' person camera viewpoints), some features will correspond to the camera support and person or vehicle carrying the camera. Also, for a camera rig that contains a nadir hole, there may be a consistent missing region in each image. These features will not move relative to the camera in the same way as the fixed scene and provide spurious correspondences. While it is possible to segment features into different motion clusters, we suggest a much simpler heuristic. The position of such points in the spherical images will be approximately fixed. Hence, in lines 3-7 and 21-25 we filter points by removing any whose position between images is closer than a threshold (we use $t_2 = 0.995$).

Nonlinear refinement and global bundle adjustment requires optimisation of (7). Note that this is not a nonlinear least squares problem like in the perspective case. Hence, we use a trust-region algorithm as implemented in the `fminunc` Matlab function.

V. Experimental Results

We now present experimental results for pose estimation, SFM and view stabilisation. We begin by solving the spherical-n-point problem on synthetic data, allowing us to evaluate the effect of noise on the camera pose estimates. Second, we evaluate the complete SFM pipeline on two real world datasets for which ground truth camera trajectories are available. Finally, we provide qualitative evaluation of stabilising a real world spherical video sequence.

[1] We follow [24] and only retain matches where the ratio between first and second nearest neighbour distances is less than a threshold, i.e. we require that $\|\mathbf{d}_{a1} - \mathbf{d}_{b2}\| / \|\mathbf{d}_{a1} - \mathbf{d}_{c2}\| < t_1$ where $\mathbf{d}_{a1}$ is the SSIFT descriptor for spherical point $\mathbf{x}_{a1}$ and $\mathbf{d}_{b2}$ and $\mathbf{d}_{c2}$ are the first and second nearest neighbours of $\mathbf{d}_{a1}$ respectively.

TABLE I: Quantitative spherical-n-point results for minimal ($n = 6$ points) setting.

κ	Method	$\sigma = 30$		$\sigma = 10$		$\sigma = 5$		$\sigma = 1$		$\sigma = 0.1$		$\sigma = 0$	
		$\varepsilon_{\text{position}}$	$\varepsilon_{\text{rotation}}$	$\varepsilon_{\text{position}}$	$\varepsilon_{\text{rotation}}$	$\varepsilon_{\text{position}}$	$\varepsilon_{\text{rotation}}$	$\varepsilon_{\text{position}}$	$\varepsilon_{\text{rotation}}$	$\varepsilon_{\text{position}}$	$\varepsilon_{\text{rotation}}$	$\varepsilon_{\text{position}}$	$\varepsilon_{\text{rotation}}$
10	S	407.89	1.08	4207.06	0.90	149.41	0.91	324.14	0.87	778.94	0.90	16614	0.90
	SW	115.28	1.59	766.35	1.35	549.76	1.34	520.64	1.31	384.95	1.37	380.47	1.68
	H	**112.28**	**0.98**	**111.63**	**0.81**	168.05	**0.84**	**189.25**	**0.80**	**223.59**	**0.82**	224.02	**0.82**
	HW	145.02	1.05	112.81	0.83	**132.80**	0.86	230.35	0.83	232.88	0.83	**165.82**	0.83
	D	541.59	1.69	479.26	1.65	534.14	1.61	432.28	1.62	479.52	1.58	745.64	1.64
50	S	812.36	0.75	100.74	0.56	158.73	0.53	**74.79**	0.49	261.48	**0.51**	142.01	**0.50**
	SW	327.38	1.18	212.97	0.86	200.96	0.78	130.33	0.67	232.02	0.80	125.29	0.87
	H	**98.89**	**0.69**	**77.65**	**0.54**	**68.98**	**0.52**	89.51	**0.48**	**66.23**	**0.51**	**73.21**	**0.50**
	HW	163.31	0.70	79.43	0.55	69.94	0.53	99.97	0.49	75.85	0.52	73.31	0.52
	D	649.60	1.44	382.72	1.17	379.30	1.06	332.70	1.09	263.65	1.07	336.14	1.12
200	S	112.62	0.65	66.79	0.40	93.27	0.34	53.75	0.30	50.19	0.30	67.38	0.30
	SW	217.38	1.01	60.36	0.55	84.88	0.45	183.96	0.37	49.21	0.39	71.31	0.42
	H	**81.05**	**0.62**	**49.58**	**0.39**	**54.86**	**0.33**	74.04	**0.29**	**37.94**	**0.30**	**41.11**	**0.29**
	HW	84.94	0.64	53.79	0.41	57.78	0.35	**73.50**	0.30	39.81	0.32	41.21	0.31
	D	464.02	1.34	234.97	0.94	291.13	0.77	233.67	0.72	205.95	0.70	273.37	0.72
500	S	211.08	0.64	57.10	0.36	31.65	0.24	28.71	**0.19**	25.94	**0.20**	32.53	**0.21**
	SW	203.39	1.05	193.72	0.47	30.92	0.30	28.07	0.21	34.87	0.24	33.69	0.24
	H	**89.37**	**0.61**	**52.93**	**0.36**	**30.85**	**0.24**	**27.64**	**0.19**	**25.38**	**0.20**	**28.56**	**0.20**
	HW	93.34	**0.61**	53.36	**0.36**	31.59	0.25	28.35	0.21	26.42	0.21	29.11	0.22
	D	553.33	1.33	223.50	0.80	180.75	0.65	150.24	0.48	105.36	0.47	105.13	0.51
1,000	S	150.86	0.63	81.79	**0.30**	37.28	0.25	24.39	0.16	20.64	**0.14**	**25.00**	**0.15**
	SW	276.97	0.97	75.53	0.40	32.05	0.24	21.00	0.17	21.56	0.15	57.38	0.18
	H	**86.47**	**0.61**	**38.27**	**0.30**	**28.11**	**0.23**	**20.95**	**0.15**	**20.28**	**0.14**	31.22	**0.15**
	HW	86.81	**0.61**	38.91	**0.30**	28.22	0.24	21.11	0.16	21.51	0.15	32.47	0.16
	D	602.07	1.32	349.10	0.76	216.20	0.46	77.57	0.41	242.82	0.42	121.73	0.37
∞	S	145.76	1.22	77.10	0.35	29.23	0.18	5.21	**0.04**	0.48	**0.004**	**0**	**0**
	SW	**16.81**	1.35	**21.80**	2.16	**19.71**	1.96	5.30	0.06	0.49	**0.004**	**0**	**0**
	H	83.52	**0.61**	38.05	**0.31**	25.62	**0.18**	**5.17**	**0.04**	**0.48**	**0.004**	**0**	**0**
	HW	86.07	0.62	41.56	0.32	26.10	0.19	5.24	**0.04**	0.51	0.0042	**0**	**0**
	D	390.45	1.33	185.37	0.70	79.72	0.37	35.76	0.09	0.94	0.0087	**0**	**0**

We vary spherical image point noise (κ is the concentration parameter of von Mises-Fisher noise) and Gaussian noise on the 3D point positions (σ is the standard deviation of the noise). The first value is the Euclidean distance between actual and estimated camera centres. The second value is the geodesic distance between actual and estimated camera rotation matrices.

A. Experimental Setup

We quantitatively evaluate camera pose estimates on synthetic data and for image sets with known ground truth pose. To do so, we use the following performance metrics.

To measure the accuracy of camera rotation, we compute the geodesic distance in the space of 3D rotations [17] between the ground truth and estimated rotation matrices. The 3D rotation group form a compact Lie group $SO(3)$ which has a natural Riemannian metric. From this, the notion of geodesic distance between two rotation matrices $\mathbf{\Omega_1}$ and $\mathbf{\Omega_2}$ follows:

$$d_g(\mathbf{\Omega_1}, \mathbf{\Omega_2}) = \left\| \log\left(\mathbf{\Omega_1} \mathbf{\Omega_2}^T \right) \right\|, \tag{26}$$

where $\log(\mathbf{\Omega})$ is the logarithmic map of $\mathbf{\Omega}$ from $SO(3)$ to $so(3)$ (i.e. the transformation to axis-angle representation). Hence, the error measure amounts to the rotation angle of the rotation matrix $\mathbf{\Omega_1}\mathbf{\Omega_2}^T$, which is in radians and is zero if $\mathbf{\Omega_1} = \mathbf{\Omega_2}$ and in general is bounded: $d_g(\mathbf{\Omega_1}, \mathbf{\Omega_2}) \in [0, \pi]$. We define the rotation error as $\varepsilon_{\text{rotation}} = d_g(\mathbf{\Omega}_{\text{groundtruth}}, \mathbf{\Omega}_{\text{estimated}})$.

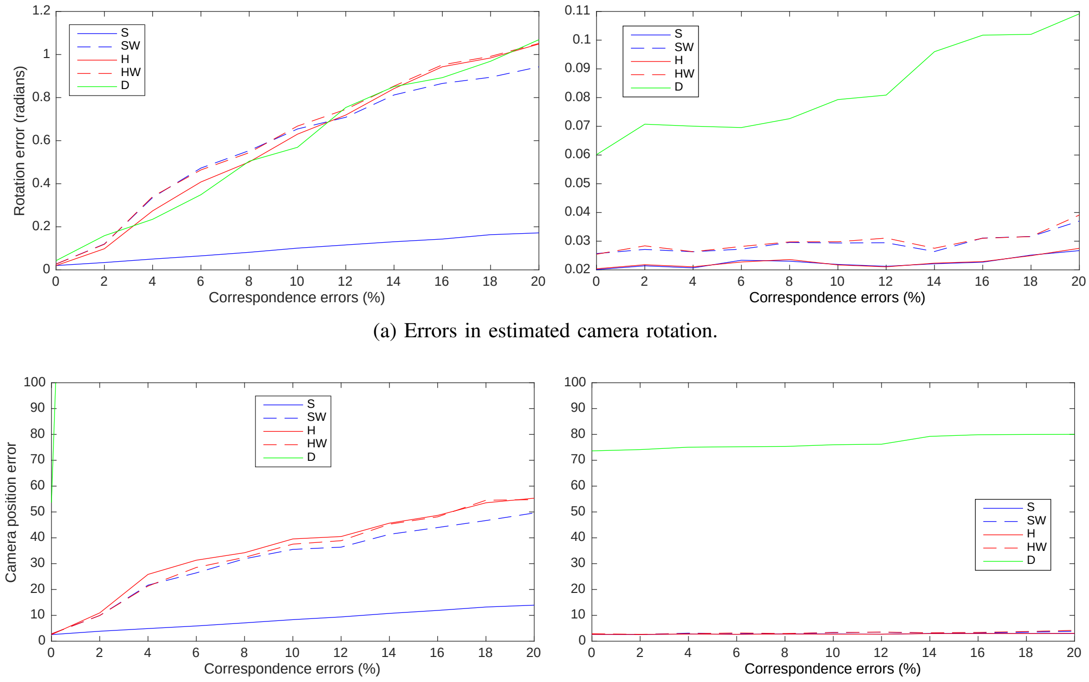
To measure the positional accuracy, we compute the implied camera centre in world coordinates from the estimated translation: $\mathbf{c} = -\mathbf{\Omega}^T \boldsymbol{\tau}$ and compute the Euclidian distance $\varepsilon_{\text{position}} = \|\mathbf{c}_{\text{groundtruth}} - \mathbf{c}_{\text{estimated}}\|$.

B. Spherical-n-point Problem

We evaluate the alternative methods we propose for solving the SnP problem using synthetic data. We randomly generate a camera rotation (by sampling uniformly from axis-angle space) and centre (by sampling from $\mathcal{N}(0, 10^2)$ for c_x, c_y and c_z). We then randomly generate n 3D points (by sampling from $\mathcal{N}(0, 100^2)$ for u, v and w). We add Gaussian noise to the 3D point positions (sampled from $\mathcal{N}(0, \sigma^2)$), project the 3D points to the virtual spherical camera using (1) and finally add vMF noise to the spherical image points. To do so, for each point we randomly sample from the vMF distribution given in (4) using the method described by Wood [38]. Every experimental configuration is repeated 1,000 times and the results averaged.

We evaluate five different methods. We refer to the classical DLT method as D. The method we propose in Section IV-C1 using a hard constraint is referred to as H or as HW when we employ the reweighting scheme in Section IV-C3. Similarly, the soft constraint in Section IV-C2 is referred to as S and as SW for the reweighted variant.

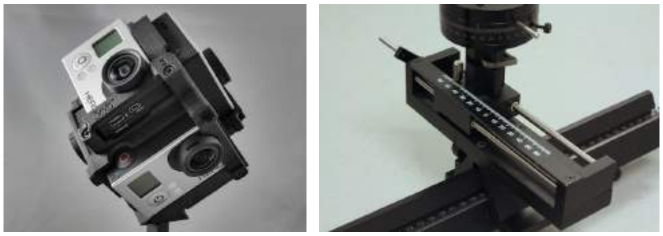
We consider two common scenarios for the SnP problem. The first is for a minimal ($n = 6$) set of points. This scenario arises when, for example, RANSAC is used to fit to noisy sets of points where each random sampling selects a minimal subset of the data. We show results for this scenario in Table I. We vary the value of the concentration parameter of von Mises-Fisher noise over the set: $\kappa = \{10, 50, 200, 500, 1,000, \infty\}$. This noise corresponds to mean angular errors in the spherical point positions of 22.8°, 10.3°, 5.13°, 3.18°, 2.26° and 0° respectively. We vary the 3D point position noise over the



(a) Errors in estimated camera rotation.

(b) Errors in estimated camera centres.

Fig. 1: Quantitative SnP results with correspondence errors. We fix the noise ($\kappa = 200$, $\sigma = 2$) and no. points ($n = 100$) and vary the no. of random correspondence errors. Left: using all points, right: RANSAC. See text for method abbreviations.



(a) Freedom360 camera rig. (b) Linear optical rail and mount.

Fig. 2: Equipment for ground truth sequence capture.

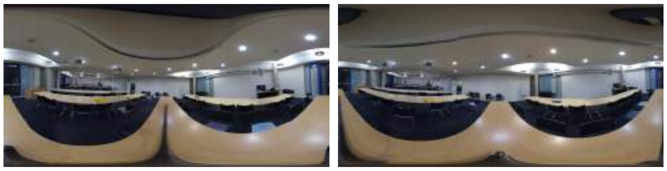


Fig. 3: Sample images from the linear trajectory dataset shown as equirectangular images.

set: $\sigma = \{30, 10, 5, 1, 0.1, 0\}$. We have emboldened the best rotation and position result for each noise setting. First, notice that all of our proposed variants outperform the classical DLT over all noise settings. Second, the weighted variants typically perform worse than the unweighted versions. This is caused by poor estimates of the weights using the noisy pose estimate from the previous iteration. As the algorithm converges, the weights do not necessarily converge to better estimates of the camera-point distances. Finally, it is clear that the unweighted hard constraint yields the most consistent performance, both in terms of the rotation and camera position accuracy. Hence, it is this method that we use in the SFM results that follow.

The second scenario we evaluate is fitting to a large set of points (in this case $n = 100$) which contains correspondence errors. To simulate correspondence errors, we take a subset (whose size is varied between 0% and 20%) of the points and randomly permute the 3D-spherical correspondences. Results are shown in Figure 1. On the left we use all points, testing resilience to outliers. Here, a different picture emerges. In the presence of correspondence errors, it is clear that the unweighted soft constraint yields significantly more robust performance. The classical DLT performs about as well as our other proposed methods in terms of rotation accuracy but is much worse in terms of camera position estimation. On the right, we use RANSAC in conjunction with each method to remove outliers. In this case, all our variants significantly outperform the DLT.

C. Quantitative Structure-from-motion Results

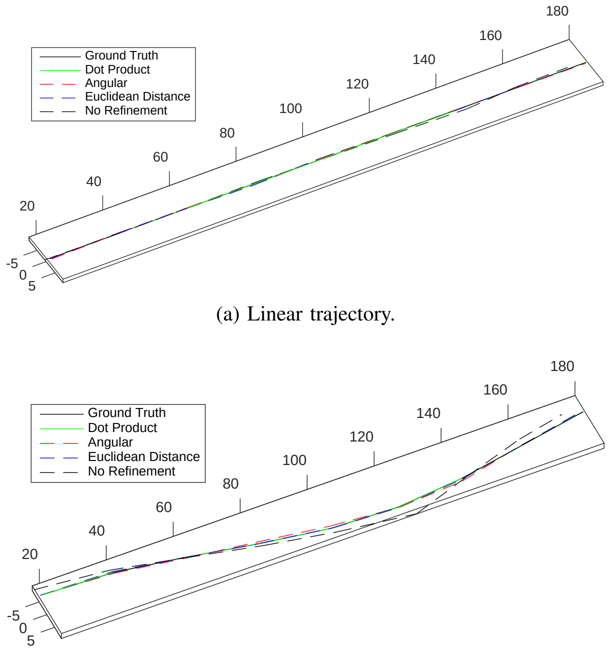
Quantitative evaluation of SFM on real image sequences is difficult since accurate ground truth of 3D world positions is hard to obtain. However, with calibrated camera motion, we can evaluate the accuracy of the camera motion trajectory estimated by our structure from motion pipeline. For this experiment, we use a Freedom360 spherical mount containing 6 GoPro Hero3 Black cameras (see Figure 2a). When the 6 images are stitched together, this provides full 180° by 360° images with no nadir blind spot.

We attach the camera rig to a mount which allows both calibrated rotation and side-to-side translation and then attach this to a linear optical rail (see Figure 2b). This allows calibrated translation in the u-w plane and rotation about the

TABLE II: Quantitative results: trajectory estimation.

Objective function	Linear trajectory		Curved trajectory	
	$\varepsilon_{\text{position}}$	$\varepsilon_{\text{rotation}}$	$\varepsilon_{\text{position}}$	$\varepsilon_{\text{rotation}}$
Squared Euclidean distance	0.67	0.0098	0.69	0.0093
Squared angular error	0.32	0.0066	0.66	0.0085
Dot product	**0.24**	**0.0030**	**0.21**	**0.0041**
No refinement	2.25	0.0119	2.75	0.0096

Errors are mean Euclidean distance between camera centres in centimetres followed by rotation error in radians.



(a) Linear trajectory.

(b) Curved trajectory.

Fig. 4: Ground truth and estimated camera trajectories for real world image sequence (zoom for detail). Results are shown without optimisation and with optimisation of three different objective functions. Distance units are centimetres.

v axis. We acquire two sequences. The first comprises a linear motion trajectory over 1.6m in the z direction in increments of 20cm. The second follows a scaled sine curve of the form $u = a\sin(bw)$ with increments of 20cm in the w direction.

We run our complete SFM pipeline on the resulting image sequences. We show two sample images from the linear trajectory sequence in Figure 3 which are visualised using an equirectangular projection (i.e. latitude and longitude mapped linearly to vertical and horizontal coordinates respectively). We evaluate four variants of the algorithm. The first applies no nonlinear refinement. In other words, we simply solve the convex SnP and calibrated spherical reconstruction problems for each new image and perform no nonlinear optimisation or bundle adjustment. The second and third methods minimise objectives used in previous work, namely the squared angular error [18] and Euclidean distance [19] between spherical image points and projected world points. Finally, our proposed method maximises likelihood under the vMF distribution, amounting to maximisation of a sum of dot products (7).

The ground truth and estimated trajectories are plotted in Figure 4 and quantitative results are shown in Table II. The errors shown are mean Euclidean distance between ground truth and estimated camera centres (in centimetres) after Procrustes alignment of the estimated trajectory to the ground truth. Maximisation of the probabilistic objective function provides the best results on both datasets.

D. Qualitative View Stabilisation Results

Finally, we provide qualitative results for our target application of stabilising spherical video. We use a video sequence of a skier descending a piste captured using the same rig as in Figure 2a.[2] The sequence is "third person" in that the camera rig is mounted on a monopod attached to the backpack of the skier (hence, the rig moves with the skier but does not turn with his head, as in a first person view). We show qualitative results from a portion of this sequence in Figures 5 and 6. In the selected frames, the skier makes a 180° turn to the left and tilts left whilst doing so.

Hence, in Figure 5 it is evident in the raw frames from the sequence (shown in the left column) that the environment is moving (note the position of the mountain peak that starts left of centre and the change in the shape of the horizon). On the other hand, the skier (who is approximately fixed relative to the camera) remains in the same position. In the stabilised frames (shown in the right hand column), we have rotated each frame back to the pose of the first frame in the sequence using the rotation matrix estimated by our spherical SFM pipeline. The effect is that distant points (whose direction remains approximately constant in the world coordinate system) are stabilised to an approximately constant position.

To further illustrate this, in Figure 6 we use the raw and stabilised panoramic images to render a virtual pinhole (perspective) view facing along the positive z-axis (roughly the direction of travel) and with a horizontal and vertical field of view of 116° and 83° respectively. This corresponds to the sort of view that may be produced when interactively viewing 360 videos. The effect of stabilisation is now very clear.

In supplementary material we include videos of the raw/stabilised panoramic/perspective views allowing a better appreciation of the stabilisation result. We also include results on different sequences. The effect of viewing the stabilised sequence as a video is of following the same trajectory as in the original but with viewing direction remaining fixed.

VI. Conclusions

In this paper we have presented a framework for stabilising the viewing direction in spherical video sequences. In doing so, we described a spherical SFM algorithm that incorporates a well justified spherical noise model (the von Mises-Fisher distribution) which leads to an objective function that is both cheaper to evaluate and performs better than the commonly used squared angular error. We have also presented constrained and weighted versions of the spherical-n-point and calibrated spherical reconstruction problems that outperform classical DLT-based approaches under a wide range of noise settings. Applying the whole pipeline to challenging real world videos yields high quality stabilisation results.

[2] Video courtesy of: Ignacio Ferrando, Abaco Digital (www.abaco-digital.es).

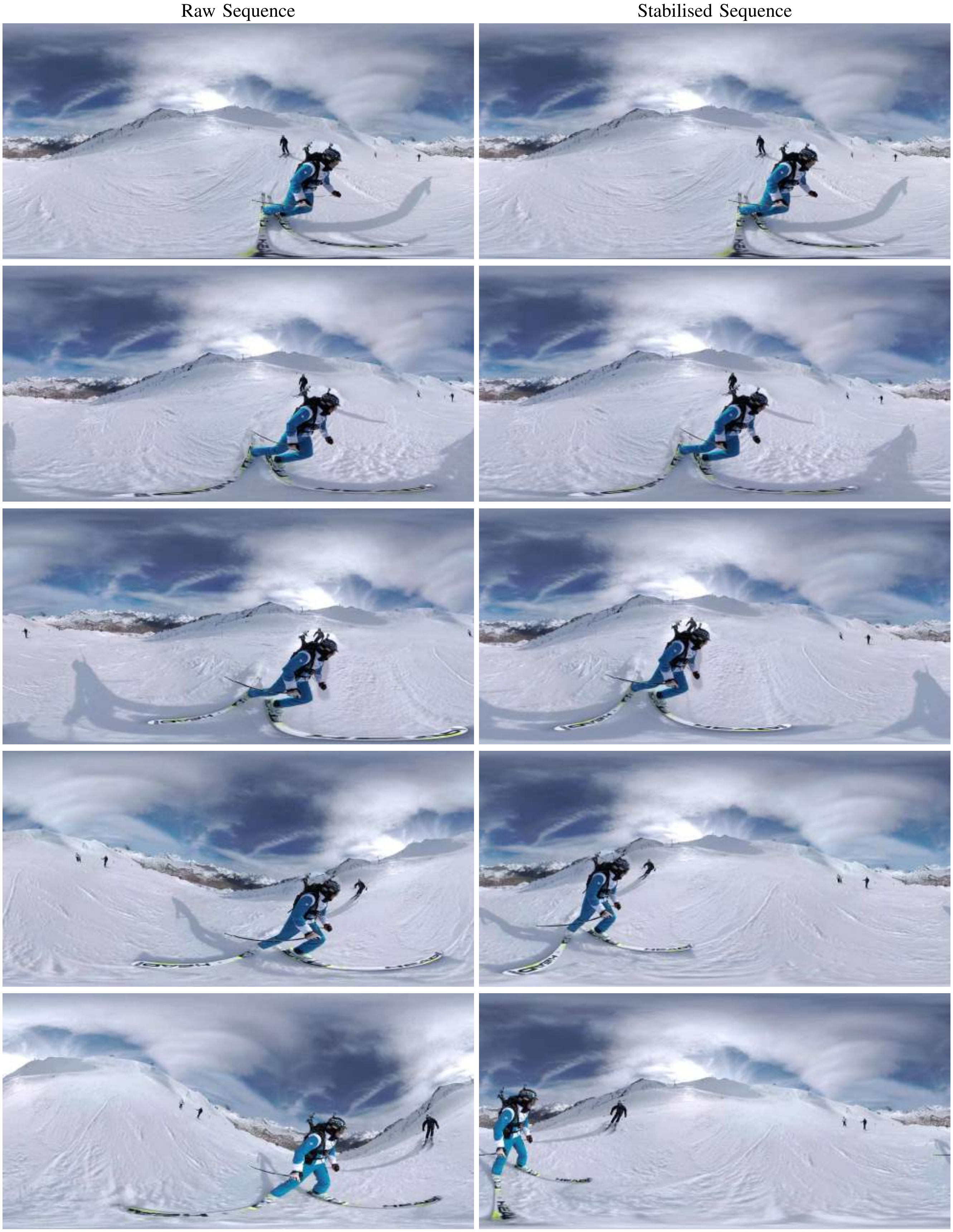


Fig. 5: Raw panoramic frames from spherical video sequence (left) and stabilised frames (right).

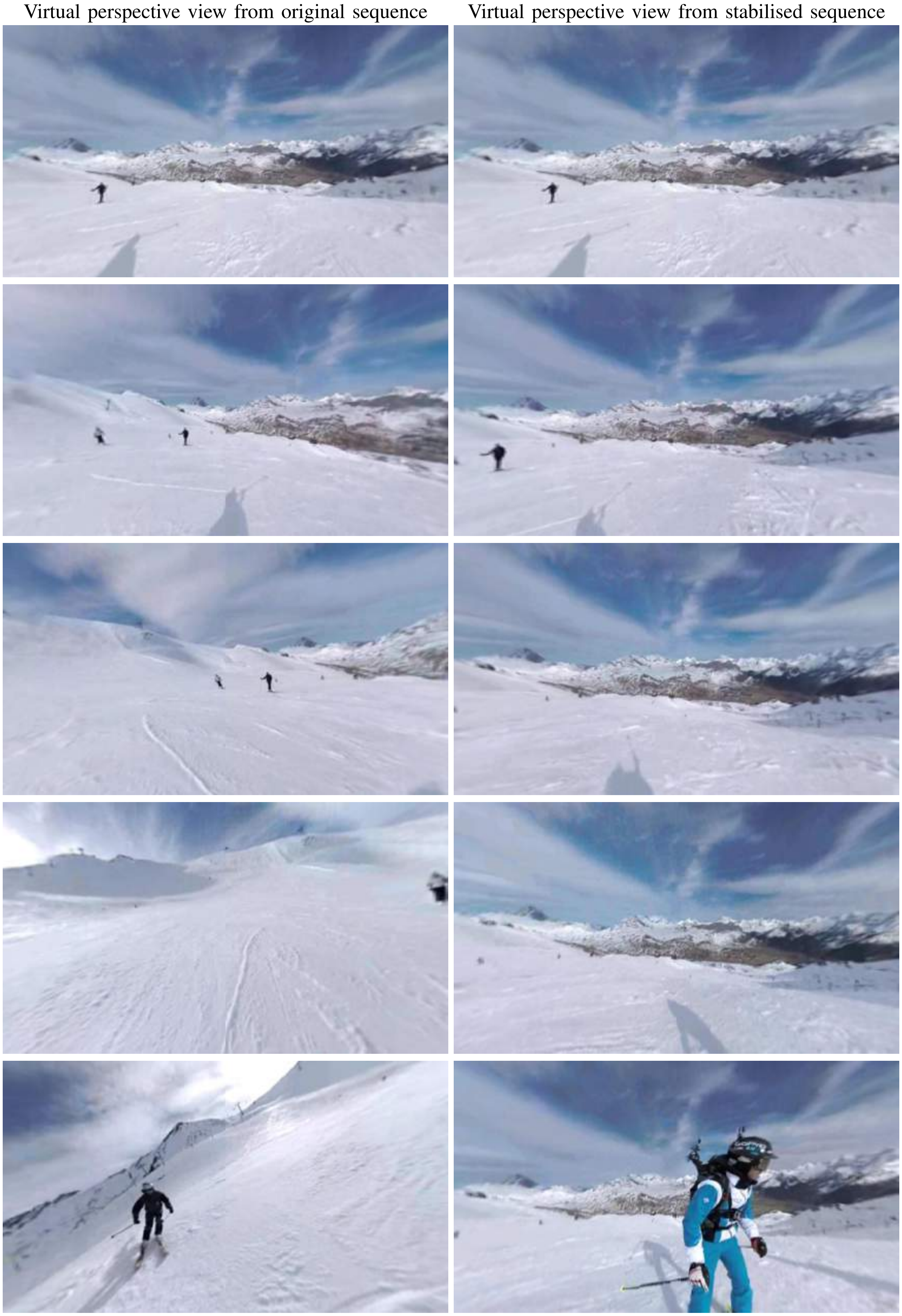

Virtual perspective view from original sequence
Virtual perspective view from stabilised sequence

Fig. 6: Virtual perspective view from raw spherical video sequence (left) and stabilised sequence (right).

There are many areas for future work. First, we would like to explicitly cluster points into those moving with the camera and those fixed in the world. This is important for first or third person video sequences. Second, we would like to conduct perceptual experiments to verify that the stabilised sequences provide a better experience when viewed by humans via a head mounted display. Third, we would like to explore viewpoint interpolation in more detail and investigate whether the mesh-warping based stabilisation algorithms that have proven effect for perspective images can be extended to the spherical case. This would allow both viewpoint stabilisation but also other applications such as spherical hyper-lapse or free viewpoint video from motion sequences.

REFERENCES

[1] Y. I. Abdel-Aziz and H. M. Karara, "Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry," in *Proc. Symp. Close-Range Photogrammetry*, 1971.

[2] J. Bai, A. Agarwala, M. Agrawala, and R. Ramamoorthi, "User-assisted video stabilization," *Computer Graphics Forum*, vol. 33, no. 4, 2014.

[3] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using von Mises-Fisher distributions," in *Journal of Machine Learning Research*, 2005, pp. 1345–1382.

[4] Y. Bastanlar, A. Temizel, Y. Yardimci, and P. Sturm, "Multi-view structure-from-motion for hybrid camera scenarios," *Image and Vision Computing*, vol. 30, no. 8, pp. 557–572, 2012.

[5] J. C. Bazin, K.-J. Yoon, I. Kweon, C. Demonceaux, and P. Vasseur, "Particle filter approach adapted to catadioptric images for target tracking application," in *Proc. BMVC*, 2009.

[6] J. Ćesić, I. Marković, and I. Petrović, "Moving objects tracking on the unit sphere using a multiple-camera system on a mobile robot," in *Proc. Intelligent Autonomous Systems*, 2016, pp. 899–911.

[7] P. Chang and M. Hebert, "Omni-directional structure from motion," in *Proc. OMNIVIS*, 2000, pp. 127–133.

[8] J. Cruz-Mota, I. Bogdanova, B. Paquier, M. Bierlaire, and J.-P. Thiran, "Scale invariant feature transform on the sphere: Theory and applications," *Int. J. Comput. Vis.*, vol. 98, no. 2, pp. 217–241, 2012.

[9] N. I. Fisher, T. Lewis, and B. J. J. Embleton, *Statistical Analysis of Spherical Data*. Cambridge University Press, 1987.

[10] C. C. Gava and D. Stricker, "SPHERA - a unifying structure from motion framework for central projection cameras," in *Proc. VISAPP*, 2015.

[11] M. L. Gleicher and F. Liu, "Re-cinematography: Improving the camera dynamics of casual video," in *Proc. 15th International Conference on Multimedia*, 2007, pp. 27–36.

[12] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110.

[13] ——, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[14] H. Guan and W. A. P. Smith, "Corner detection in spherical images via accelerated segment test on a geodesic grid," in *Proc. International Symposium on Visual Computing*, 2013, pp. 407–415.

[15] D. Gutierrez, A. Rituerto, J. Montiel, and J. J. Guerrero, "Adapting a real-time monocular visual slam from conventional to omnidirectional cameras," in *Proc. OMNIVIS*, 2011, pp. 343–350.

[16] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[17] D. Q. Huynh, "Metrics for 3D rotations: Comparison and analysis," *J. Math. Imaging Vis.*, vol. 35, no. 2, pp. 155–164, 2009.

[18] M. Kamali, A. Banno, J. C. Bazin, I. S. Kweon, and K. Ikeuchi, "Stabilizing omnidirectional videos using 3D structure and spherical image warping," in *IAPR Conf. MVA*, 2011, pp. 177–180.

[19] B. Krolla, C. Gava, A. Pagani, and D. Stricker, "Consistent pose uncertainty estimation for spherical cameras," in *Proc. Int. Conf. Computer Graphics, Visualization and Computer Vision*, 2014.

[20] M. Lhuillier, "Effective and generic structure from motion using angular error," in *Proc. ICPR*, vol. 1, 2006, pp. 67–70.

[21] F. Liu, M. L. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3D video stabilization," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 44:1–44:9, 2009.

[22] F. Liu, M. L. Gleicher, J. Wang, H. Jin, and A. Agarwala, "Subspace video stabilization," *ACM Trans. Graph.*, vol. 30, no. 1, 2011.

[23] S. Liu, L. Yuan, P. Tan, and J. Sun, "Bundled camera paths for video stabilization," *ACM Trans. Graph.*, vol. 32, no. 4, 2013.

[24] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[25] I. Markovic, F. Chaumette, and I. Petrovic, "Moving object detection, tracking and following using an omnidirectional camera on a mobile robot," in *Proc. ICRA*, 2014, pp. 5630–5635.

[26] B. Mičušík and T. Pajdla, "Para-catadioptric camera auto-calibration from epipolar geometry," in *Proc. ACCV*, 2004, pp. 748–753.

[27] A. C. Murillo, D. Gutiérrez-Gómez, A. Rituerto, L. Puig, and J. J. Guerrero, "Wearable omnidirectional vision system for personal localization and guidance," in *2nd IEEE Workshop on Egocentric (First-Person) Vision*, 2012, pp. 8–14.

[28] F. Nielsen, "Surround video: a multihead camera approach," *The Visual Computer*, vol. 21, no. 1-2, pp. 92–103, 2005.

[29] A. Pagani and D. Stricker, "Structure from motion using full spherical panoramic cameras," in *Proc. OMNIVIS*, 2011.

[30] A. Pagani, C. Gava, Y. Cui, B. Krolla, J.-M. Hengen, and D. Stricker, "Dense 3d point cloud generation from multiple high-resolution spherical images," in *Proc. International Conference on Virtual Reality, Archaeology and Cultural Heritage (VAST)*, 2011, pp. 17–24.

[31] G. Pons-Moll, A. Baak, J. Gall, L. Leal-Taix, M. Mueller, H.-P. Seidel, and B. Rosenhahn, "Outdoor human motion capture using inverse kinematics and von Mises-Fisher sampling," in *Proc. ICCV*, 2011, pp. 1243–1250.

[32] B. Schmeing, T. Läbe, and W. Förstner, "Trajectory reconstruction using long sequences of digital images from an omnidirectional camera," *DGPF Tagungsband*, 2011.

[33] P. Sturm and S. Ramalingam, "A generic concept for camera calibration," in *Proc. ECCV*, 2004, pp. 1–13.

[34] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera," in *Proc. IROS*, 2008, pp. 2531–2538.

[35] A. Torii, I. Atsushi, and O. Naoya, "Two-and three-view geometry for spherical cameras," in *Proc. OMNIVIS*, 2005.

[36] A. Torii, M. Havlena, and T. Pajdla, "From Google street view to 3D city models," in *Proc. OMNIVIS*, 2009, pp. 2188–2195.

[37] M. Vosmeer and B. Schouten, "Interactive cinema: engagement and interaction," in *Interactive Storytelling*. Springer, 2014, pp. 140–147.

[38] A. T. A. Wood, "Simulation of the von Mises Fisher distribution," *Commun. Stat-Simul. Comp.*, vol. 23, no. 1, pp. 157–164, 1994.

[39] Q. Zhao, W. Feng, L. Wan, and J. Zhang, "SPHORB: A fast and robust binary feature on the sphere," *International Journal of Computer Vision*, vol. 113, no. 2, pp. 143–159, 2015.

Hao Guan received the B.Sc. degree in Software Design and Development from Nanchang University, China in 2009 and the M.Sc. degree in Computing with Vision and Imaging from the University of Dundee, UK in 2011. He is currently pursuing the Ph.D. degree at the University of York where he is a member of the Computer Vision and Pattern Recognition research group. His research interests include spherical video processing, structure-from-motion and computer vision.

William A. P. Smith (M'08) received the B.Sc. degree in computer science, and the Ph.D. degree in computer vision from the University of York, York, U.K. He is currently a Senior Lecturer with the Department of Computer Science, University of York, York, U.K. His research interests are in shape and appearance modelling and physics-based vision. He has published more than 90 papers in international conferences and journals