Robust Frame-to-Frame Camera Rotation Estimation in Crowded Scenes

Fabien Delattre [1] David Dirnfeld [1] Phat Nguyen [1] Stephen Scarano [1]
Michael J. Jones [2] Pedro Miraldo [2] Erik Learned-Miller [1]

[1] University of Massachusetts Amherst [2] Mitsubishi Electric Research Laboratories (MERL)

Abstract

We present an approach to estimating camera rotation in crowded, real-world scenes from handheld monocular video. While camera rotation estimation is a well-studied problem, no previous methods exhibit both high accuracy and acceptable speed in this setting. Because the setting is not addressed well by other datasets, we provide a new dataset and benchmark, with high-accuracy, rigorously verified ground truth, on 17 video sequences. Methods developed for wide baseline stereo (e.g., 5-point methods) perform poorly on monocular video. On the other hand, methods used in autonomous driving (e.g., SLAM) leverage specific sensor setups, specific motion models, or local optimization strategies (lagging batch processing) and do not generalize well to handheld video. Finally, for dynamic scenes, commonly used robustification techniques like RANSAC require large numbers of iterations, and become prohibitively slow. We introduce a novel generalization of the Hough transform on $SO(3)$ to efficiently and robustly find the camera rotation most compatible with optical flow. Among comparably fast methods, ours reduces error by almost 50% over the next best, and is more accurate than any method, irrespective of speed. This represents a strong new performance point for crowded scenes, an important setting for computer vision. The code and the dataset are available at https://fabiendelattre.com/robust-rotation-estimation.

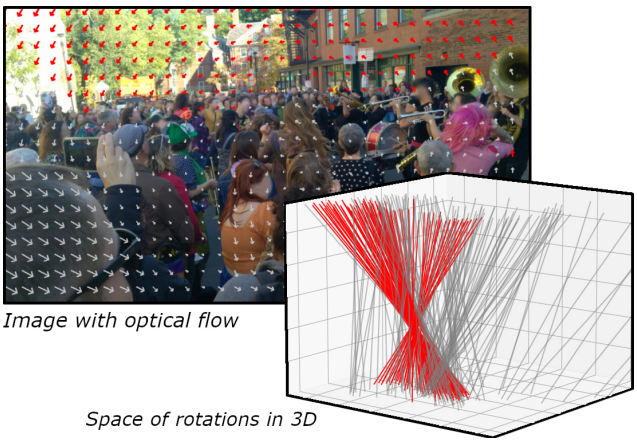

Figure 1. **Left.** A frame from our BUSS dataset of crowded scenes. The red vectors show optical flows compatible with the winning rotation estimate R^*, indicating the rotation of the camera. Gray vectors show optical flows not explained purely by R^*. **Right.** The three axes show the space of rotations in 3D. Each line shows the one-dimensional set of rotations that are compatible with a single optical flow vector. The red lines (corresponding to the red flow vectors in the top figure) intersect in a single small bin, indicating that their optical flows are compatible with the same rotation. The gray lines, which are affected by other motion effects, are scattered in an unstructured manner, and correspond to the gray optical flows above. Our algorithm finds the set of lines with greatest coherence in $SO(3)$, revealing the rotation R^* of the camera.

1 Introduction

The estimation of camera motion through a scene is a fundamental problem in computer vision that is highly related to a number of vision tasks such as motion segmentation [5], video stabilization [44], 3D reconstruction [9], visual odometry [56], Simultaneous Localisation and Mapping (SLAM) [53], Structure-from-Motion (SfM) [65], human-computer interaction [54], autonomous navigation [69], and many more. Hence, developing a method that can accurately predict the camera's movement through a scene is critical in solving these problems.

As the camera moves through the scene, the motion field depends not only on the camera's motion but also on the scene's geometry and objects' motion in the environment. Given a sufficiently crowded location with many moving objects (e.g., pedestrians and vehicles), estimating the camera motion requires the difficult task of distinguishing between static and moving objects. This paper proposes a novel, robust method of estimating camera rotation in crowded scenes such as the one shown in Fig. **1**.

It is important to clarify the difference between frame-

to-frame camera motion estimation and relative pose estimation. Specifically, camera motion estimation is a constrained version of relative pose estimation where only two views are used, constrained to be (a) spatially close, (b) temporally close, and (c) taken from the same camera, which matches the case of adjacent frames in a moving-camera video.

Nowadays, many authors focus on relative pose estimation using point correspondences. Most of these methods focus on estimating the essential matrix [45, 56], which works best in the presence of large parallax [47, Remark 5.2] (large baselines). Therefore, correspondence-based methods are primarily used for offline localization and mapping strategies such as SfM and 3D reconstruction, or online pipelines with local optimization like SLAM. In contrast, optical flow-based methods are better suited for small motions, which is the domain of interest in this paper.

As in state-of-the-art correspondence-based relative pose problems [34], the best optical flow-based methods for frame-to-frame camera motion estimation focus on decoupling the transformation into rotation- and translation-only estimation [5,7]. While there are fast and accurate solutions to motion estimation, they are highly sensitive to moving objects in the scene–they frequently break down with significant numbers of moving objects in the scene. Similarly to correspondence-based techniques, optical flow-based methods are often used within RANSAC [14] to handle locally wrong optical flow and moving objects, and thereby increase robustness. In this paper, we focus on rotation estimation since flow-based translation estimation given rotation estimates can be easily computed as shown in [5, 7].

We propose a new method to estimate the camera rotation based on optical flow. Our approach can be used for highly dynamic scenes, from the assumption that optical flow from faraway points is less sensitive to dynamic objects in the scene. The proposed technique uses a compatible rotation voting mechanism and does not require RANSAC (see Fig. 1). In addition, since public datasets only contain static scenes or have minor dynamic objects (a large portion of the frames contain static environments), we acquire a new and challenging dataset of 17 sequences in (anonymized) crowded environments. The dataset will be made available. To summarize, our contributions are as follows:

- A novel robust frame-to-frame camera rotation estimation algorithm based on optical flow that finds compatible rotations using a voting mechanism based on the Hough transform in the space of 3D rotations;
- We show that our algorithm significantly outperforms the discrete and continual baselines in highly dynamic scenes and performs comparably in static scenes; and
- We provide a new dataset of highly dynamic scenes called BUsy Street Scenes (BUSS) that comes with rigorously verified ground truth rotation.

2 Related work

Motion estimation methods can be classified into three groups: differential methods, discrete methods, and direct methods. Differential methods model the pixel displacements between two frames as instantaneous 3D velocities, while discrete methods model the pixel displacements as 3D translations and rotations. Direct methods typically avoid defining displacements explicitly, and are based on brightness-constancy constraints. Our method can be used in either the differential or discrete paradigms.

Differential methods: These methods [62] (also known as instantaneous-time [72] or continual methods) use visual motion field for estimating camera motion, and are thereby well-suited for small motions. We start by reviewing methods based on the motion model proposed by Longuet-Higgins and Prazdny [46].

In [7, 48], the authors formulate the problem to be independent of scene depth, and solve using nonlinear numerical optimization using a weighted bilinear constraint. Kanatani [29] shows that these methods introduce bias. To remove the bias, Zhang and Tomasi [75] propose an iterative method with an unweighted bilinear constraint that optimizes for translation using the Gauss-Newton method. To avoid local minima of previous methods, Pauwels and Van Hulle [58] start with the weighted bilinear constraint and gradually move to the unweighted bilinear constraint. The authors in [61, 63] leverage the fact that in areas of depth discontinuities, the difference of flows is due to translation. Inspired by this work, Heeger proposes multiple subspace methods [19, 20, 25] that also solve for translation first. The difference is that the solution is not an approximation and the sampled flow vectors do not need to be close to each other.

In [59], Perrone tunes flow vector location-specific detectors to respond to different translation or rotation directions and speeds. He then adopts a voting scheme to determine the best-fitting rotation and translation. A drawback is that this approach requires a huge number of templates to cover the 5 continuous dimensions. In a subsequent work, Perrone [60] proposes to first stabilize the gaze, reducing the number of dimensions to optimize. Lappe [39] proposes a biologically plausible implementation of [19].

Methods in [30, 51, 77] use a differential version of the epipolar constraint. [30, 77] propose algorithms to linearly solve for the continual fundamental matrix.

Discrete methods: These methods do not make assumptions about frame-to-frame displacements, e.g., [4, 31, 43, 50, 73]. The literature is vast. We list a few key works.

Most discrete methods use the epipolar constraint, [12,

24, 45, 47], and can be split into two groups: calibrated and uncalibrated ones. In both, most authors focus on deriving minimal solvers for RANSAC. In the calibrated case, the essential matrix can be estimated using 5-point correspondences (see [3, 18, 38, 42, 52, 55]). Some authors proposed other minimal solvers for improving speed[1]: [28] derives 6- and 7-point solvers, and a DLT method (8-point algorithm) is presented in [47]. [41] uses $SE(3)$ invariances for constraining the minimal solvers. Others were proposed for constrained motions, such as [40, 64]. For the uncalibrated cases, again many authors proposed different solvers. In [17], the author proposes an 8-point algorithm. [37, 70] propose minimal 6-point algorithms for solving the relative pose with an unknown common focal length. [11] explores the use of IMU for deriving a 4-point algorithm. In [36], the authors study problems with radial distortion.

Some authors focused on offering non-minimal solvers for fine estimates, such as [6, 10, 76]. In [34, 35], the authors propose a new epipolar constraint based on the coplanarity of epipolar plane normal vectors. [8] avoids possible local minima by using an estimate of an unsupervised pose network. In [15], the authors use a robust loss function to detect and discard outliers.

Direct methods: Instead of explicitly computing the optical flow, direct methods solve for camera motion using the brightness-consistency constraint equation (e.g., [13, 21, 67, 68, 74]). Despite spatial-temporal gradient information, no closed-form solution exists, and strong assumptions must be made to simplify the problem. Horn and Weldon proposed several algorithms for cases of pure rotation, pure translation, or when depth is known [22]. Some works have shown that even when there exist rotation and translation, a solution can be found by considering the world as planer [1, 22, 26], or by enforcing chirality constraint (the depth remains positive) [2, 57]. Direct methods suffer from changes of illumination, and they become extremely slow when run using a robust estimator framework to handle moving objects.

Robust motion estimation: To handle moving objects and noise in the (dense or sparse) correspondences, motion estimation methods are usually run within robust estimators. Bideau et al. [5] develops a loss function to evaluate the quality of camera rotations with respect to optical flows, and perform gradient descent of this function in $SO(3)$. Unfortunately, local minima of the loss can be caused by similarity of rotation flows to translation and moving object flows, leading to poor solutions. To avoid this, one may do exhaustive search, such as in [59], by discretizing rotation space into C_{rot} 3D bins, and evaluate every rotation in time $\mathcal{O}(C_{rot})$. In general, large bins yield poor accuracy and smaller bins are too computationally expensive. Intuitively, this approach wastes large amounts of computation on very poor rotation candidates. RANSAC [14] takes a different approach in which a random sample of flow vectors are evaluated for consistency with a particular motion model. This method works well when there are few outliers. However, with large numbers of outliers, such as in crowded street scenes, the required number of RANSAC iterations becomes too large, yielding very slow algorithms (see run time results for RANSAC algorithms). Thus, RANSAC is not viable when the percentage of outlier pixels is large. To tackle those challenges, we propose a generalization of the Hough Transform on $SO(3)$ that we describe in Sec. **3.2**.

[1]The 5-point solver estimates up to 10 solutions for each sampling hypothesis, requiring up to 10 inlier counting per hypothesis. Non-minimal solvers get a single solution per hypothesis.

3 Proposed approach

Our goal is to estimate the camera rotation between two frames, given $\{u_i, v_i, x_i, y_i\}$ where (u_i, v_i) are optical flow vectors and (x_i, y_i) are their respective coordinates in the image plane. Consider an optical flow field F caused purely by camera rotation, with no camera translation, moving objects, or noise. As we discuss in Sec. **3.1**, each flow vector in such a rotation field provides two constraints on the set of possible rotations, as shown in Fig. **1**. For a purely rotational optical flow field, the lines intersect in a single point, the rotation that causes the optical flow.

However, in real-world video, optical flow is also affected by translation, moving objects and noise. In general, there exists no single rotation compatible with all the optical flow vectors. To estimate the rotation, we leverage the fact that the flows of distant points are mostly affected by rotation and thus behave approximately like 'rotation-only' flow vectors. The hypothesis is that these distant points will provide consistent evidence for a particular rotation, while other flow vectors, influenced by translation, scene geometry, moving objects, and noise, will not produce a consistent estimate of rotation. Thus, by accumulating evidence (or votes) for the rotation with strongest support, we can estimate the camera rotation.

Of course, this highlights an important assumption of our method: we assume that camera translations between frames are small relative to distant points in the scene. This ensures that the flows of distant scene points are well-modeled by rotations. Thus, our method is designed to work in outdoor scenes (or spacious indoor scenes, like arenas) where the translational camera motions are small relative to the most distant objects.

Our method can be considered a variation of the well-known Hough transform [23]. The Hough transform attempts to find the hidden variable that could have generated as many observations as possible. Each observation is used to "vote" for the hidden variable values with which it is consistent. In our case, the observations are optical flow vectors

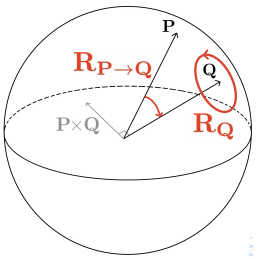

Figure 2. **Retrieving the set of rotations mapping P to Q**. The set of all rotations that map P to Q (a one-dimensional submanifold of $SO(3)$) can be obtained by composing $\mathbf{R}_{P\to Q}$, a single rotation, with $\mathbf{R}_Q$, the set of all rotations about the vector Q.

(at each point in the image), and the hidden variable values are the possible rotations. This approach can be considered a "robustification" method, since it allows us to get good estimates in the presence of large numbers of "outliers", i.e., flows influenced by other factors (translation, moving objects, poor optical flow estimates).

The rest of this section is as follows: We review the perspective projection motion model and the Longuet-Higgins motion model in Sec. **3.1.1** and Sec. **3.1.2**, and we derive the set of compatible rotations using both models; In Sec. **3.2**, we introduce our Hough transform based voting scheme and compare the computational efficiency of our method to other robust methods.

3.1 Compatible rotations

In this section, we discuss how to find the set of rotations that can produce a specific optical flow vector that is only affected by camera rotation. Given that the space of 3D rotations $SO(3)$ is a 3D manifold (rotations about the 3 axes) and that optical flow vectors have two degrees of freedom (u and v), there is a one-dimensional set of rotations with which any flow vector is compatible. We present two versions of our method, a discrete version using perspective projection, and a continual version using the Longuet-Higgins motion model.

3.1.1 The perspective projection motion model

In this section we review classical materials on perspective projection, and we show how to compute the set of rotations that can produce a particular flow vector under perspective projection. Consider a camera, aligned with world coordinates, that images a point with world coordinates P and image coordinates $\mathbf{p}$. Now consider a rotation of the camera so that the world point in the new camera frame is given by camera coordinates Q and image location $\mathbf{q}$. Because the magnitude of P and Q are the same (rotations do not change vector magnitudes), and the magnitudes of P and Q do not affect their projections onto the image, we can assume they both have unit magnitude.

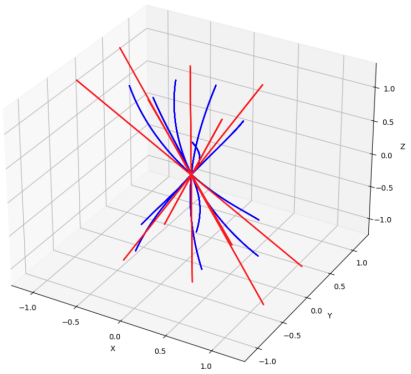

Figure 3. **Longuet-Higgins vs. perspective projection.** Each flow vector is compatible with a 1D manifold of rotations (axes in radians). Here, we show (partial) sets of compatible rotations using the Longuet-Higgins model (straight red lines) and (partial) sets of compatible rotations using perspective projection (blue curves.) Our algorithm can be used with either motion model.

The set of all rotations that map P to Q (a one-dimensional submanifold of $SO(3)$) can be obtained by composing $\mathbf{R}_{P\to Q}$, a single rotation about $P \times Q$, with $\mathbf{R}_Q^\theta$, a rotation about the vector Q by an angle θ that does not change the position of Q in the camera's frame. This can be done for any angle θ, generating a one-dimensional manifold of rotations (see Fig. **2**) that can be written as

$$\mathcal{R}(P,Q) = \{\mathbf{R} : \mathbf{R} = \mathbf{R}_Q^\theta \mathbf{R}_{P\to Q}, 0 \le \theta < 2\pi\}.$$

Given a set of rotations that can map P to Q, we next discuss how to find rotations that map $\mathbf{p}$ to $\mathbf{q}$ in an image (i.e., the rotations that are compatible with optical flow vector $\mathbf{q}-\mathbf{p}$). First, to get P and Q, we take the inverse images of $\mathbf{p}$ and $\mathbf{q}$ (assuming that P and Q are unit vectors). The set of rotations compatible with the flow vector $\mathbf{q}-\mathbf{p}$ is therefore $\mathcal{R}(P,Q)$. To compute this from P and Q we define $\mathbf{R}_{P\to Q}$, which consists of an axis of rotation $P \times Q$, and the angle of rotation given by $\arccos(P \cdot Q)$. For $\mathbf{R}_Q^\theta$, the axis of rotation is simply Q, and the angle of rotation is any θ such that $0 \le \theta < 2\pi$. This one-dimensional family of rotations $\mathcal{R}(P,Q)$ is a curve in $SO(3)$ (blue curves in Fig. **3**). Next, we show how the Longuet-Higgins model yields a slightly different set of compatible rotations.

3.1.2 Using Longuet-Higgins motion model

The Longuet-Higgins visual motion field model for static scenes [46] defines an instantaneous motion field velocity (rate of change of position in the image) as

$$\mathbf{v} = \underbrace{\begin{pmatrix} \frac{A}{f}xy - Bf - \frac{B}{f}x^2 + Cy \\ Af + \frac{A}{f}y^2 - \frac{B}{f}xy - Cx \end{pmatrix}}_{\mathbf{v_r}} + \underbrace{\begin{pmatrix} \frac{-fU+xW}{Z} \\ \frac{-fV+yW}{Z} \end{pmatrix}}_{\mathbf{v_t}}. \quad (1)$$

The motion field velocity $\mathbf{v}$ is represented as a sum of the 2D rotational velocity $\mathbf{v_r}$ and the 2D translational velocity $\mathbf{v_t}$. These in turn are defined as functions of the 3D translational velocities U, V, W, the 3D rotational velocities A, B, C, the depth Z, the image positions x, y and the focal length f. For motion caused only by rotation, we have, for a specific image location (x, y),

$$\mathbf{v}(x,y) = \begin{pmatrix} A\left(\frac{xy}{f}\right) - B\left(\frac{f^2+x^2}{f}\right) + Cy \\ A\left(\frac{f^2+y^2}{f}\right) - B\left(\frac{xy}{f}\right) - Cx \end{pmatrix}. \quad (2)$$

These equations lead to a 1D manifold of solutions, a line l at the intersection of two planes defined by the two equations in Eq. 2. **The simple form of this 1D manifold (a straight line) allows a very fast implementation of the Hough transform, as described in Sec. 3.2.**

Let $\mathbf{n}_u$ and $\mathbf{n}_v$ be normal vectors to these planes:

$$\mathbf{n_u} = \left[\frac{xy}{f}, -\frac{f^2+x^2}{f}, y\right], \ \mathbf{n_v} = \left[\frac{f^2+y^2}{f}, -\frac{xy}{f}, -x\right]. \quad (3)$$

The line l defined by the intersection of the two planes has direction $\mathbf{d} = \mathbf{n_u} \times \mathbf{n_v}$. By simple algebra, it can be shown that the z component of $\mathbf{d}$ can't be 0, which implies that the line l can't be co-planar to the plane $C = 0$. Therefore, we can complete the definition of l by finding its intersection with the plane $C = 0$, by setting $C = 0$ in Eq. 2. Notice that the direction of l (given by the vector $\mathbf{d}$) is independent of the optical flow vector. Only the intercept depends on the flow vector. Therefore, one can precompute line directions for each image location, and only find the intercept at run time, resulting in major efficiency gains. Figure 3 shows the 1D manifolds of compatible rotations produced by perspective projection and the Longuet-Higgins motion model. A comparison of the accuracy and the run time of the two approaches can be found in Sec. 6. In the subsequent sections of the paper, we will report results of our method using the Longuet-Higgins motion model.

3.2 Voting Scheme

We discretize the 1D manifold of solutions we get from Sec. 3.1 into rotation votes. Unlike the original Hough transform, we do not create an accumulator, but make a list of compatible rotation votes, and find the mode of the list, alleviating the need for a 3-dimensional accumulator in memory. In summary, our approach allows dense sampling of $SO(3)$ while maintaining rapid execution. Our method's speed depends on the number of flow vectors C_{OF} used in voting and the number of points sampled per 1D manifold of compatible rotations. We sample about $\sqrt[3]{C_{rot}}$ points per 1D manifold, the approximate number of bins intersected by each line. Thus, our total number of 'votes' and hence our complexity is $\mathcal{O}(C_{OF}\sqrt[3]{C_{rot}})$.

Figure 4. **BUSS dataset**. Example frames from our BUSS dataset. The sequences are recorded in different scenes and have a diverse set of camera motion.

Table 1. **Dataset comparison**. Comparison of our proposed BUSS dataset to other relevant datasets.

	BUSS (ours)	IRSTV [66]	Cambridge Landmarks [32]	KITTI [16]
Year	2022	2021	2016	2012
Platform	Hand held	Hand held	Hand held	Car
Scene	Outdoors	In/outdoors	Outdoors	Outdoors
% of moving objects	Very high	Very low	Low	Very low
Anno. freq.	30Hz	20Hz	2Hz	10Hz
Rot. GT	IMU	IMU	SfM	IMU
Baseline	Small	Small	Large	Large
Num. frames	5,504	7,800	10,929	43,552

4 Dataset

We introduce *BUsy Street Scenes* (BUSS), a challenging dataset of video sequences taken from a handheld mobile phone (an OPPO A5 2020 smartphone, rear camera) in crowded city streets with synchronized inertial measurement unit (IMU) data. The goal of the dataset is to evaluate the robustness of camera rotation estimation algorithms in dense and dynamic scenes with many moving objects and complex camera motion. The dataset composes 17 video sequences of about 10 seconds each at 30fps in full HD resolution (1920x1080) RGB. We used the *Android Open-Camera Sensor* app to synchronously record video and angular rate from the phone's MEMS gyroscopes (at 400Hz) and then generated the rotation ground truth using the method we discuss in Sec. 4.1. To meet strict privacy standards, videos are only captured in public places, and faces and other personally identifiable information (PII) is blurred. Along with the anonymized video frames, we also provide optical flow for all sequences computed with RAFT [71]. All sequences show highly dynamic scenes (see Fig. 4).

4.1 Ground truth calculation

The BUSS ground truth was estimated using the angular rate measurements recorded simultaneously with the video. The ground truth rotation at frame f_t represents the for-

ward rotation from the video frame f_t to the immediate next frame f_{t+1}. To get the rotation between two frames, we numerically integrate angular rate measurements [49].

To assess the reliability of our dataset's rotational ground truth, we compared the measurements of the OPPO with the measurements of a different phone (iPhone 12 mini) with the phones bound to the same rigid surface. Comparing two gyroscope sensor models gives us strong confidence in data correctness since it is highly unlikely that the two phones (with different hardware) agree on erroneous measurements. After recording gyroscope data simultaneously from both phones, we corrected for temporal and spatial misalignment. We synchronized the internal clocks of the two phones by finding the time offset that minimized disagreement error. For spatial misalignment, we corrected for the relative orientation $\mathbf{R}$ between the two gyroscopes using the Kabsch algorithm on the rotation velocity vectors [27]. The average frame-to-frame (at 30 fps) rotation error between the two phones is $0.014°$. This is an order of magnitude smaller than the errors from state-of-the-art methods in a similar setting. This validates the choice of using a gyroscope to generate the ground truth of the dataset. See additional details in supplementary material.

4.2 Comparison to existing datasets

Our proposed BUSS dataset has three key properties that are not found simultaneously in any publicly available dataset: (a) it is recorded with a handheld camera, introducing highly variable camera motions (b) it contains highly dynamic scenes, and (c) it has high frequency, accurate and synchronized rotation ground truth. The IRSTV dataset [66] does not have property (b) because the number of moving objects is sparse. The Cambridge Landmarks dataset [32] contains some sequences with dynamic scenes, but the ground truth rotations are only given at 2 FPS. The popular KITTI dataset [16] has few moving vehicles and pedestrians per frame and the camera is mounted on a vehicle, so the dataset is lacking all three properties. The comparison of the main characteristics of our dataset with three other publicly available datasets can be found in Tab. **1**.

5 Experiment

We evaluate our method for frame-to-frame rotation estimation on our proposed BUSS dataset and on IRSTV [66]. The properties of both datasets are described in Sec. **4.2**.

5.1 Evaluation Metrics

To evaluate the rotation estimation accuracy, we use Average Angular Error (AAE) which computes the angle of rotation between the ground-truth rotation and estimated rotation. Let $f_{i,j}$ be the j^{th} frame in the i^{th} sequence. Let $\mathbf{R}_{i,j}$ be the ground-truth rotation between frame $f_{i,j}$ and frame $f_{i,j+1}$, and $\hat{\mathbf{R}}_{i,j}$ be the estimated rotation between the same two frames. Then, the average angular error is given by

$$AAE = \frac{1}{\sum_{i=1}^{N} M_i} \sum_{i=1}^{N} \sum_{j=1}^{M_i} \theta(\hat{\mathbf{R}}_{i,j} \mathbf{R}_{i,j}^{-1}), \tag{4}$$

where N is the number of sequences, M_i is the number of optical flows for the sequence i and $\theta(\cdot)$ is the magnitude of rotation resulting from $\hat{\mathbf{R}}_{i,j} \mathbf{R}_{i,j}^{-1}$.

5.2 Implementation Details

All methods, including ours, are run on an Intel Xeon CPU. For the continual baselines, we used the MATLAB implementations and the RANSAC parameters provided in [62]. For the discrete baselines, we used the implementations and the RANSAC parameters from OpenGV [33]. To offer a fair run time comparison against other continual baselines, our method is also implemented in MATLAB. We use a bin size of 0.057 degrees (see the ablation in Sec. **6**), and we search over rotations between -4 and 4 degrees.

For the methods that require the optical flow, including ours, we first resize the video frames to a size of 480x270. We then compute the optical flow using RAFT [71] on a GeForce GTX 1080Ti GPU. RAFT offers a good trade-off between performance and speed. We then resized the optical flow to 32x18 by sampling the optical flow vectors on a regular grid with stride 15. An ablation on the spatial sampling rate can be found in Sec. **6**.

For the methods based on feature correspondences, we extract 3000 SIFT descriptors from the full-resolution 1920x1080 video frames. We match them using a standard brute-force matcher.

5.3 Results

We compare the frame-to-frame rotation estimation from our Longuet-Higgins method against several continual baselines: Bruss&Horn (B&H) [7], Heeger&Jepson (H&J) [20], Kanatani (Kan) [30], Lappe&Rauschecker (L&R) [39], Pauwels&Van Hulle (P&V) [58], Zhang&Tomasi (Z&T) [75], and discrete baselines: Kneip (Kne) [34] and Nistér also known as the 5-points algorithm (Nis) [55]. In addition of running the continual methods using all the optical flow vectors, we also run all continual methods, except Lappe&Rauschecker [39] and ours which are robust to moving objects by design, in RANSAC for 1, 25, 100 and 500 RANSAC iterations. For the discrete baselines, we ran each method for 500, 5000 and 50000 iterations. To quantify the amount of rotation in both datasets, we also include the zero baseline as reference.

Results on BUSS: The results on the BUSS dataset clearly illustrate the strength of our approach. Table **2** reports the

numerical results, and Fig. 5 shows the rotation error vs. run time. Our method is almost 50% more accurate than comparably fast methods. Due to the highly dynamic nature of the BUSS dataset, RANSAC significantly improves the accuracy of the other methods (ranging from 66% to 30% improvement). Yet, even with the improvement gained from RANSAC, our method outperforms the second-best method by 25% while being more than 400 times faster. The standard error of the mean is smaller than 1.3% for our method, and smaller than 7% for the other methods.

	Rotation err. (°)					Time per frame (seconds)				
Continual methods										
# iters.	N/A	1	25	100	500	N/A	1	25	100	500
B&H [7]	0.21	0.28	0.17	—*	—*	0.14	9.92	226.01	—*	—*
H&J [20]	0.25	0.40	0.21	0.18	0.16	0.13	0.26	3.30	10.72	62.09
Kan [30]	0.28	0.60	0.24	0.21	0.20	0.12	0.23	2.86	8.37	37.54
L&R [39]	0.30	—	—	—	—	13.07	—	—	—	—
P&V [58]	0.22	0.30	0.22	0.21	0.21	0.12	0.61	5.62	13.27	38.76
Z&T [75]	0.22	0.30	0.22	0.21	0.21	0.13	0.51	4.93	11.82	36.50
Ours	0.12	—	—	—	—	0.14	—	—	—	—
Discrete methods										
# iters.		500	5K	50K			500	5K	50K	
Kneip [34]		0.69	0.36	0.33			1.63	3.00	6.04	
Nistér [55]		0.43	0.34	0.33			1.86	3.26	10.91	

Table 2. **Quantitative results on the BUSS dataset**. We compare the frame-to-frame rotation error and the run time of our method with multiple other baselines for multiple number of RANSAC iterations. Experiments where the number of iterations is "N/A" means that the experiments have been run without RANSAC.
**The run time is too long to run the experiment.*

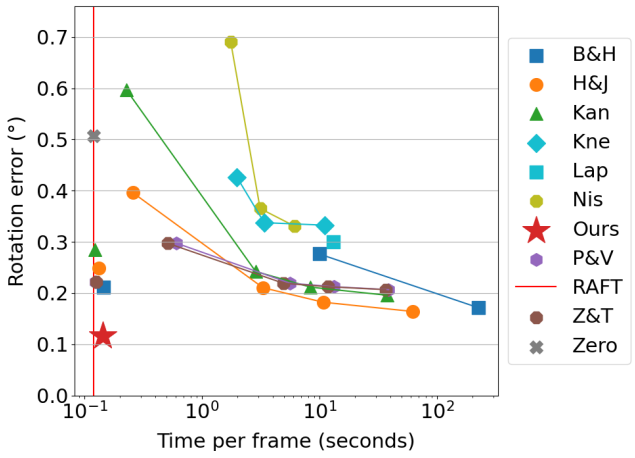

Figure 5. **Rotation error vs. run time on BUSS.** Methods run with RANSAC appear on a line, with different numbers of RANSAC iterations at each point. Standalone points do not use RANSAC. The run time of the continual methods includes the run time of the optical flow computation.

Results on IRSTV: The results for the IRSTV dataset are reported in Tab. 3. We show the plot of the rotation error vs. run time in Fig. 6. Our method is on par with the other methods with respect to accuracy and speed. Our method has a rotation error of 0.14° while operating at 0.15 seconds per frame. Due to the fact that IRSTV is mostly composed of static scenes, running continual methods with RANSAC only marginally improve the results while increasing the run time significantly.

	Rotation err. (°)					Time per frame (seconds)				
Continual methods										
# iters.	N/A	1	25	100	500	N/A	1	25	100	500
B&H [7]	0.12	0.19	0.13	—*	—*	0.15	9.15	238.45	—*	—*
H&J [20]	0.15	0.40	0.18	0.14	0.12	0.13	0.26	3.28	11.09	59.64
Kan [30]	0.15	0.34	0.14	0.13	0.12	0.12	0.22	2.32	10.11	43.04
L&R [39]	0.14	—	—	—	—	12.78	—	—	—	—
P&V [58]	0.15	0.20	0.15	0.14	0.14	0.12	0.51	5.48	17.09	76.47
Z&T [75]	0.15	0.20	0.15	0.14	0.14	0.12	0.51	5.36	17.95	78.69
Ours	0.14	—	—	—	—	0.15	—	—	—	—
Discrete methods										
# iters.		500	5K	50K			500	5K	50K	
Kne [34]		0.28	0.27	0.26			1.64	2.57	4.82	
Nis [55]		0.33	0.30	0.30			1.56	2.02	2.77	

Table 3. **Quantitative results on IRSTV dataset**. We compare the frame-to-frame rotation error and the run time of our method with multiple other baselines for multiple number of RANSAC iterations. Experiments where the number of iterations is "N/A" means that the experiments have been run without RANSAC.
**The run time is too long to run the experiment.*

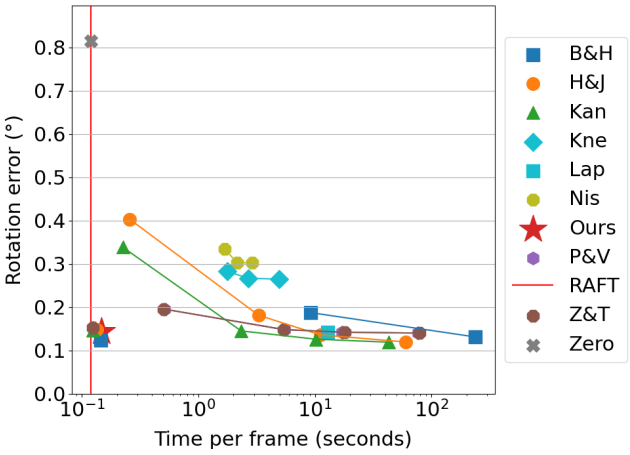

Figure 6. **Rotation error vs. run time on IRSTV.** Methods connect by lines use RANSAC. Standalone points do not.

The results on IRSTV and BUSS show the robustness of our method to moving objects. While the rotation error of our proposed algorithm stays comparable across the two datasets, the rotation errors of the baselines increase on the BUSS dataset. It's worth noting that continual methods perform significantly better than the discrete methods

on both datasets, which suggests that discrete methods are more susceptible to noise. Additionally, the Zero baseline error is more significant ($\approx 0.8°$) on the IRSTV dataset than on the BUSS dataset ($\approx 0.5°$) due to IRSTV having a lower frame rate. This also explains why our method performs slightly worse on IRSTV than on BUSS.

5.4 Robustness to moving and close objects

In this section, we investigate the proportion of pixels in the frames needed to be far away. In the case of pure rotation, the winning rotation bin receives votes from all flow vectors. Fig. 7 shows the percentage of flow vectors in the winning bin for the BUSS dataset. 62% of the optical flows have less than 25% of the flow vectors in the winning bin, with the majority resulting in small errors (<0.2deg). This shows that our algorithm is highly effective even when most of the flow vectors are affected by a translation or by moving objects.

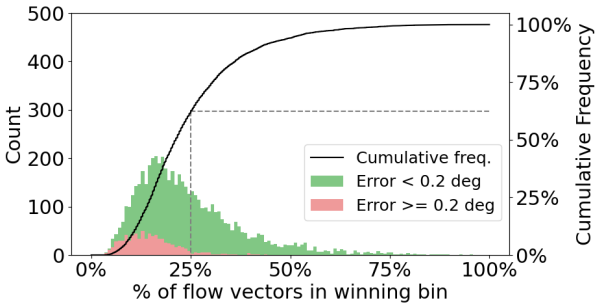

Figure 7. Percentage of flow vectors that has voted for the winning rotation bin on the BUSS dataset. E.g., the dotted line shows that for 62% of the optical flows, less than 25% of the flow vectors are in the winning bin.

6 Ablation Studies

We compare motion models (perspective vs. Longuet-Higgins), different quantizations of rotation space, and the spatial sampling rates of optical flow.

Varying bin sizes on $SO(3)$: Fig. 8 compares rotation error and run time for our two approaches on BUSS. Both methods demonstrate similar rotation accuracy regardless of bin size. However Longuet-Higgins is much faster for small bins. There is a sweet spot for bin size. If bins are too small, noise in optical flow prevents the 1d manifold of compatible rotations from intersecting the correct bin. But making bins bigger increases error when picking the correct bin. Since the rotation estimate is the bin's center, the maximum error when choosing the correct bin is $\left(\sqrt{3}/2\right) s$, for bin size s.

Robustness to spatial sampling: We sample optical flow vectors along a regular grid. Fig. 9 shows our model's robustness to spatial sampling step sizes on BUSS. The rotation error remains between 0.11 degrees and 0.13 degrees for step sizes ranging from 1 to 80. This allows subsampling optical flow and reducing run time. When flow sampling becomes too sparse, rotation error increases due to overexposure to potentially noisy flow vectors. Surprisingly, rotation error also slightly increases when flow sampling becomes too dense. We hypothesize that this is because far away points are spatially distributed on the frames, while objects, that could potentially exhibit motions that are coherent to a rotation, are generally well-bounded in space. Therefore, after diminishing the sampling rate, we still benefit from far away points across the frame, while reducing flows sampled from the same object.

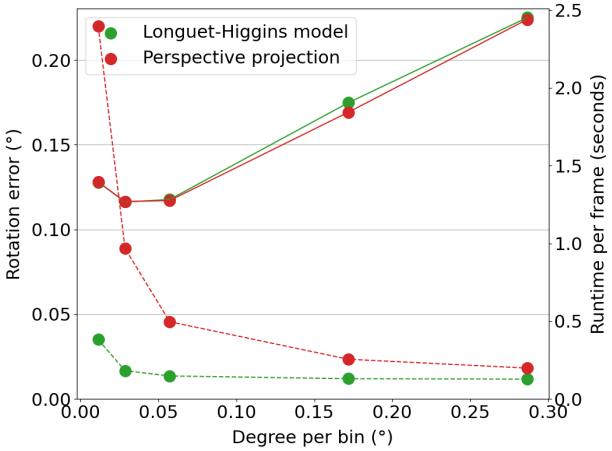

Figure 8. **Performance on BUSS as a function of bin size.** Our method's accuracy (continuous line) and run time (dashed line) with perspective projection and Longuet-Higgins. The methods have similar accuracy but Longuet-Higgins is much faster.

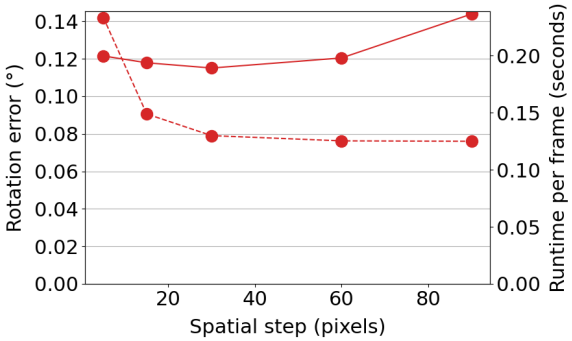

Figure 9. **Performance on BUSS as a function of spatial step size**. Our error (continuous line) and run time (dashed line) for different spatial step values. A spatial step value of n means that we sample flow vectors every n pixels.

7 Conclusion

We introduce a novel generalization of the Hough transform on $SO(3)$ to find the camera rotation most compatible with optical flow in highly dynamic scenes. Our method is inherently robust, and doesn't need RANSAC, which significantly improves the speed over existing methods. In presence of moving objects, our method reduces the error by almost 50% over the next best method for the same run time, while performing similarly in static scenes. Additionally, we propose a challenging new dataset BUSS that consists of 17 video sequences in crowded, real-world scenes.

References

[1] Mingxin Ai, Tie Liu, Haocong Ying, Zejian Yuan, Jing Wang, and Yuanyuan Shang. A direct and robust method for ego-motion estimation. *China Automation Congress (CAC)*, pages 1349–1353, 2021. 3

[2] Francisco Barranco, Cornelia Fermüller, Yiannis Aloimonos, and Eduardo Ros. Joint direct estimation of 3d geometry and 3d motion using spatio temporal gradients. *ArXiv*, abs/1805.06641, 2021. 3

[3] Dhruv Batra, Bart C. Nabbe, and Martial Hebert. An alternative formulation for five point relative pose problem. *IEEE Workshop on Motion and Video Computing (WMVC'07)*, pages 21–21, 2007. 3

[4] Luis Baumela, Lourdes Agapito, P Bustos, and I Reid. Motion estimation using the differential epipolar equation. In *IEEE Int'l Conf. Pattern Recognition (ICPR)*, volume 3, pages 840–843, 2000. 2

[5] Pia Bideau and Erik G. Learned-Miller. It's moving! a probabilistic model for causal motion segmentation in moving camera videos. In *European Conf. Computer Vision (ECCV)*, page 433–449, 2016. 1, 2, 3

[6] Jesus Briales, Laurent Kneip, and Javier González. A certifiably globally optimal solution to the non-minimal relative pose problem. *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 145–154, 2018. 3

[7] Anna R. Bruss and Berthold K. P. Horn. Passive navigation. *Computer Vision, Graphics, and Image Processing*, 20(1):3–20, 1982. 2, 6, 7

[8] Claudio Cimarelli, Hriday Bavle, José Luis Sánchez-López, and Holger Voos. Raum-vo: Rotational adjusted unsupervised monocular visual odometry. *Sensors*, 22(7):2651, 2022. 3

[9] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH*, page 11–20, 1996. 1

[10] Yaqing Ding, Daniel Barath, Jian Yang, Hui Kong, and Zuzana Kukelova. Globally optimal relative pose estimation with gravity prior. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 394–403, 2021. 3

[11] Yaqing Ding, Daniel Barath, Jian Yang, and Zuzana Kukelova. Relative pose from a calibrated and an uncalibrated smartphone image. In *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 12756–12765, 2022. 3

[12] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT press, 1993. 2

[13] Cornelia Fermüller and Yiannis Aloimonos. Qualitative egomotion. *Int'l J. Computer Vision (IJCV)*, 15:7–29, 2005. 3

[14] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2, 3

[15] Mercedes Garcia-Salguero and Javier Gonzalez-Jimenez. Fast and robust certifiable estimation of the relative pose between two calibrated cameras. *J. Mathematical Imaging and Vision*, 63(8):1036–1056, 2021. 3

[16] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 5, 6

[17] Richard I. Hartley. In defense of the eight-point algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence (T-PAMI)*, 19(6):580–593, 1997. 3

[18] Richard I. Hartley and Hongdong Li. An efficient hidden variable approach to minimal-case camera motion estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence (T-PAMI)*, 34(12):2303–2314, 2012. 3

[19] David J. Heeger and Allan D. Jepson. Visual perception of three-dimensional motion. *Neural Computation*, 2:129–137, 1990. 2

[20] David J. Heeger and Allan D. Jepson. Subspace methods for recovering rigid motion i: Algorithm and implementation. *Int'l J. Computer Vision (IJCV)*, 7:95–117, 2004. 2, 6, 7

[21] Joachim Heel. Direct estimation of structure and motion from multiple frames. 1990. 3

[22] Berthold K. P. Horn and E. J. Weldon. Direct methods for recovering motion. *International Journal of Computer Vision*, 2:51–76, 2004. 3

[23] P V.C. Hough. Method and means for recognizing complex patterns. 12 1962. 3

[24] Thomas S. Huang and Olivier D. Faugeras. Some properties of the e matrix in two-view motion estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence (T-PAMI)*, 11:1310–1312, 1989. 2

[25] Allan D. Jepson and David J. Heeger. Linear subspace methods for recovering translational direction. 1994. 2

[26] Hui Ji and Cornelia Fermuller. A 3d shape constraint on video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1018–1023, 2006. 3

[27] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32:922–923, 1976. 6

[28] F. Kahl and B. Triggs. Critical motions in euclidean structure from motion. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 366–372, 1999. 3

[29] Kenichi Kanatani. Renormalization for unbiased estimation. *1993 (4th) International Conference on Computer Vision*, pages 599–606, 1993. 2

[30] Kenichi Kanatani. 3-d interpretation of optical flow by renormalization. *Int'l J. Computer Vision (IJCV)*, 11:267–282, 2005. 2, 6, 7

[31] Kenichi Kanatani and Naoya Ohta. Comparing optimal three-dimensional reconstruction for finite motion and optical flow. *Journal of Electronic Imaging*, 12(3):478–488, 2001. 2

[32] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. 5, 6

[33] Laurent Kneip and Paul Timothy Furgale. Opengv: A unified and generalized approach to real-time calibrated geometric vision. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8, 2014. 6

[34] Laurent Kneip and Simon Lynen. Direct optimization of frame-to-frame rotation. *IEEE Int'l Conf. Computer Vision (ICCV)*, pages 2352–2359, 2013. 2, 3, 6, 7

[35] Laurent Kneip, Roland Y. Siegwart, and Marc Pollefeys. Finding the exact rotation between two images independently of the translation. In *European Conf. Computer Vision (ECCV)*, pages 696–709, 2012. 3

[36] Yubin Kuang, Jan Erik Solem, Fredrik Kahl, and Kalle Åström. Minimal solvers for relative pose with a single unknown radial distortion. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 33–40, 2014. 3

[37] Zuzana Kukelova, Martin Bujnak, and Tomas Pajdla. Automatic generator of minimal problem solvers. In *European Conf. Computer Vision (ECCV)*, pages 302–315, 2008. 3

[38] Zuzana Kukelova, Martin Bujnak, and Tomás Pajdla. Polynomial eigenvalue solutions to the 5-pt and 6-pt relative pose problems. In *British Machine Vision Conference (BMVC)*, 2008. 3

[39] Markus Lappe and Josef P. Rauschecker. A neural network for the processing of optic flow from ego-motion in man and higher mammals. *Neural Computation*, 5(3):374–391, 1993. 2, 6, 7

[40] Bo Li, Lionel Heng, Gim Hee Lee, and Marc Pollefeys. A 4-point algorithm for relative pose estimation of a calibrated camera with a known relative rotation angle. In *IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS)*, pages 1595–1601, 2013. 3

[41] Bo Li, Evgeniy Martyushev, and Gim Hee Lee. Relative pose estimation of calibrated cameras with known se(3) invariants. In *European Conf. Computer Vision (ECCV)*, page 215–231, 2020. 3

[42] Hongdong Li and Richard I. Hartley. Five-point motion estimation made easy. *IEEE Int'l Conf. Pattern Recognition (ICPR)*, 1:630–633, 2006. 3

[43] Wen-Yan Lin, Geok-Choo Tan, and Loong-Fah Cheong. When discrete meets differential: Assessing the stability of structure from small motion. *Int'l J. Computer Vision (IJCV)*, 86:87–110, 2009. 2

[44] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM Transactions on Graphics*, 32(4), 2013. 1

[45] Hugh Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981. 2

[46] Hugh Christopher Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 208(1173):385–397, 1980. 2, 4

[47] Yi Ma, Stefano Soatto, Jana Košecká, and Shankar Sastry. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer, 2004. 2, 3

[48] W. James MacLean, Allan D. Jepson, and Richard C. Frecker. Recovery of egomotion and segmentation of independent object motion using the em algorithm. In *British Machine Vision Conference (BMVC)*, 1994. 2

[49] Sebastian O. H. Madgwick, Andrew J. L. Harrison, and Ravi Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–7, 2011. 6

[50] Markus Mainberger, Andrés Bruhn, and Joachim Weickert. Is dense optic flow useful to compute the fundamental matrix? In *International Conference Image Analysis and Recognition (ICIAR)*, pages 630–639, 2008. 2

[51] Stephen J. Maybank. The angular velocity associated with the optical flowfield arising from motion through a rigid environment. *Proceedings of the Royal Society A. Mathematical, Physical and Engineering Sciences*, 401(1821):317–326, 1985. 2

[52] Stephen J. Maybank, Thomas S. Huang, Teuvo Kohonen, and Manfred R. Schroeder. Theory of reconstruction from image motion. In Stephen Maybank, editor, *Springer Series in Information Sciences*, volume 28. Springer Berlin, Heidelberg, Heidelberg, 1993. 3

[53] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Trans. Robotics (T-RO)*, 31(5):1147–1163, 2015. 1

[54] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *IEEE Int'l Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136, 2011. 1

[55] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Analysis and Machine Intelligence (T-PAMI)*, 26(6):756–770, 2004. 3, 6, 7

[56] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, volume 1, 2004. 1, 2

[57] Chethan Parameshwara, Gokul Hari, Cornelia Fermuller, Nitin J. Sanket, and Yiannis Aloimonos. Diffposenet: Direct differentiable camera pose estimation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6835–6844, 2022. 3

[58] Karl Pauwels and Marc M. Van Hulle. Optimal instantaneous rigid motion estimation insensitive to local minima. *Computer Vision and Image Understanding (CVIU)*, 104(1):77–86, 2006. 2, 6, 7

[59] John A. Perrone. Model for the computation of self-motion in biological systems. *Journal of the Optical Society of America A (JOSAA)*, 9(2):177–194, 1992. 2, 3

[60] John A. Perrone and Leland S. Stone. A model of self-motion estimation within primate extrastriate visual cortex. *Vision Research*, 34:2917–2938, 1994. 2

[61] K. Prazdny. Egomotion and relative depth map from optical flow. *Biological Cybernetics*, 36:87–102, 2004. 2

[62] Florian Raudies and Heiko Neumann. A review and evaluation of methods estimating ego-motion. *Computer Vision and Image Understanding (CVIU)*, 116(5):606–633, 2012. 2, 6

[63] Joachim H. Rieger and Daryl T. Lawton. Processing differential image motion. *Journal of the Optical Society of America A (JOSAA)*, 2(2):354–359, 1985. 2

[64] Davide Scaramuzza. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *Int'l J. Computer Vision (IJCV)*, 95:74–85, 2011. 3

[65] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 1

[66] Myriam C. J. Servieres, Valérie Renaudin, Alexis Dupuis, and Nicolas Antigny. Visual and visual-inertial slam: State of the art, classification, and experimental benchmarking. *J. Sensors*, 2021:2054828:1–2054828:26, 2021. 5, 6

[67] César Silva and José Santos-Victor. Direct egomotion estimation. *IEEE Int'l Conf. Pattern Recognition (ICPR)*, 1:702–706, 1996. 3

[68] César Silva and José Santos-Victor. Robust egomotion estimation from the normal flow using search subspaces. *IEEE Trans. Pattern Analysis and Machine Intelligence (T-PAMI)*, 19(9):1026–1034, 1997. 3

[69] Annett Stelzer, Heiko Hirschmüller, and Martin Görner. Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain. *The International Journal of Robotics Research (IJRR)*, 31(4):381–402, 2012. 1

[70] Henrik Stewenius, David Nister, Fredrik Kahl, and Frederik Schaffalitzky. A minimal solution for relative pose with unknown focal length. *Image and Vision Computing (IVC)*, 26(7):871–877, 2008. 3

[71] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conf. Computer Vision (ECCV)*, page 402–419, 2020. 5, 6

[72] Tina Yu Tian, Carlo Tomasi, and David J. Heeger. Comparison of approaches to egomotion computation. *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 315–320, 1996. 2

[73] Bill Triggs. Differential matching constraints. In *IEEE Int'l Conf. Computer Vision (ICCV)*, volume 1, pages 370–376, 1999. 2

[74] Ding Yuan and Yalong Yu. A new method on camera egomotion estimation. *International Congress on Image and Signal Processing (CISP)*, 2:651–656, 2013. 3

[75] Tong Zhang and Carlo Tomasi. Fast, robust, and consistent camera motion estimation. *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, 1:164–170 Vol. 1, 1999. 2, 6, 7

[76] Ji Zhao. An efficient solution to non-minimal case essential matrix estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence (T-PAMI)*, 44(4):1777–1792, 2022. 3

[77] Xinhua Zhuang, Thomas S. Huang, Narendra Ahuja, and Robert M. Haralick. A simplified linear optic flow-motion algorithm. *Computer Vision, Graphics, and Image Processing*, 42(3):334–344, 1988. 2