

Monocular Ego-motion Estimation with a Compact Omnidirectional Camera

Wolfgang Stürzl, Darius Burschka, and Michael Suppa

Abstract—We present a generalization of the Koenderink–van Doorn (KvD) algorithm that allows robust monocular localization with large motion between the camera frames for a wide range of optical systems including omnidirectional systems and standard perspective cameras. The KvD algorithm estimates simultaneously ego-motion parameters, i.e. rotation, translation, and object distances in an iterative way. However due to the linearization of the rotational component of optic flow, the original algorithm fails for larger rotations. We present a generalization of the algorithm to arbitrary rotations that is especially suited for omnidirectional cameras where features can be tracked for long sequences. This reduces the need for vector summation of several individual motion estimates that leads to accumulation of odometry errors.

The significant improvement in the performance of the proposed generalized algorithm compared to the original KvD implementation is validated using simulated data. The algorithm is also tested in a real-world experiment with ground-truth data obtained from an external tracking system. The experiment was carried out using a novel compact omnidirectional camera that is designed for small aerial vehicles. It consists of an off-the-shelf webcam that is combined with a reflective surface machined into acrylic glass.

I. INTRODUCTION

Ego-motion estimation is an essential task for most robotic applications. The task can be simplified if the environment and the geometric structure of the observed features are known. In many cases, this simplification does not hold and the system needs to localize in an unknown environment. In this case, not only the change in the position has to be estimated but also the geometric relations and the reliability of the landmarks observed in the images. There exist approaches in Computer Vision that allow a direct estimation of the pose change from correspondences between feature points in two frames. Known approaches are the “8-point algorithm” [1] and its variations [2], [3] that work for point sets which are not all in a plane. Homography matrix based approaches [4], [5] are used for planar cases which are common especially for flying systems observing the world from higher altitudes. Most of these approaches are not able to provide a way to weigh the contribution of each observed point in an image sequence.

There are several approaches that use omnidirectional cameras. For example, Vassallo et al. [6] estimate ego-motion using a non-central catadioptric imaging system. Since their

W. Stürzl is with the Department of Neurobiology and the Center of Excellence “Cognitive Interaction Technology”, Bielefeld University, Germany wolfgang.stuerzl@uni-bielefeld.de

D. Burschka is with the Department of Informatics, Technische Universität München, Germany burschka@cs.tum.edu

S. Suppa is with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR) Michael.Suppa@dlr.de



Fig. 1. Omnidirectional camera (highlighted by the red ellipse) on a flying robot. The USB camera is connected to a “Beagle Board” computer (see <http://beagleboard.org>) that controls the four-rotor flyer.

approach, similar to the original KvD algorithm presented in the next section, is based on linearized equations, it is not suited for estimating large rotations between frames. Using a Point Grey Ladybug camera, Tardif et al. [7] recently presented a monocular visual odometry and SFM algorithm based on key frames. To obtain accurate results without bundle adjustment, estimation of orientation is decoupled from translation estimation. Torii et al. [8] estimate the trajectory of a high resolution fish-eye camera with low frame rate (3fps) by means of wide-baseline matching using affine feature regions. Scaramuzza et al. [9] exploit the non-holonomic constraints of wheeled vehicles enabling them to estimate motion from a single point correspondence. Although highly efficient, this approach is not applicable to flying vehicles.

In contrast to the majority of algorithms for ego-motion estimation, the KvD algorithm [10] simultaneously estimates the camera motion parameters, i.e. rotation and translation, and the scene structure. Although an omnidirectional view was considered in the derivation of the original algorithm, it was proposed for estimating small motions as they are usually observed between consecutive frames from a perspective camera. However, in omnidirectional cameras – in particular during rotations around the optical axis of the camera — features remain usually visible for much longer than in standard perspective cameras. As long as corresponding features can be found in the current image and in the reference image, rotation and translation (direction) of the camera can be directly inferred instead of integrating over several successive frames. Since each motion estimate can introduce a small error, a direct computation of motion

between the reference frame and the current frame avoids an accumulation of errors. In this paper we describe an algorithm that avoids the linearizations in the KvD algorithm and allows the estimation of arbitrarily large rotations. We also present experiments with a compact omnidirectional camera that is designed for small mobile robots, in particular flying systems (Fig. 1).

II. GENERALIZATION OF THE KvD ALGORITHM

In the following, an algorithm is derived that estimates ego-motion, i.e. translation vector \mathbf{T} and rotation matrix $\hat{\mathbf{R}}$, from corresponding features in two images, and the inverse distances (called “nearnesses” in [10]) of the features $\mu_i \propto \frac{1}{r_i}$.

We describe feature directions as determined from a camera image by unit vectors on the sphere \mathbf{e}_i , $\|\mathbf{e}_i\| = 1$. Feature directions after a movement of the camera are denoted by \mathbf{e}'_i .

The 3D positions of a feature in the reference frame of the camera are related according to

$$\mathbf{x}'_i = \hat{\mathbf{R}}^{-1}(\mathbf{x}_i - \mathbf{T}), \quad (1)$$

with $\mathbf{x}_i = r_i \mathbf{e}_i$ being the position before the movement and $\mathbf{x}'_i = r'_i \mathbf{e}'_i$ being the position after the movement. Thus, for the projection on the sphere we find

$$\begin{aligned} \mathbf{e}'_i &= \frac{\mathbf{x}'_i}{\|\mathbf{x}'_i\|} = \frac{\hat{\mathbf{R}}^{-1}(\mathbf{x}_i - \mathbf{T})}{\|\hat{\mathbf{R}}^{-1}(\mathbf{x}_i - \mathbf{T})\|} = \frac{\hat{\mathbf{R}}^{-1}(r_i \mathbf{e}_i - \mathbf{T})}{\|r_i \mathbf{e}_i - \mathbf{T}\|} \\ &= \frac{\hat{\mathbf{R}}^{-1}(\mathbf{e}_i - \frac{1}{r_i} \mathbf{T})}{\|\mathbf{e}_i - \frac{1}{r_i} \mathbf{T}\|} = \frac{\hat{\mathbf{R}}^{-1}(\mathbf{e}_i - \mu_i \mathbf{T})}{\|\mathbf{e}_i - \mu_i \mathbf{T}\|} \\ &= \hat{\mathbf{R}}^{-1} \frac{\mathbf{e}_i - \mu_i \mathbf{T}}{\sqrt{1 - 2\mu_i \mathbf{e}_i^\top \mathbf{T} + \mu_i^2 \mathbf{T}^2}}, \quad (2) \end{aligned}$$

$$\iff \hat{\mathbf{R}} \mathbf{e}'_i = \frac{\mathbf{e}_i - \mu_i \mathbf{T}}{\sqrt{1 - 2\mu_i \mathbf{e}_i^\top \mathbf{T} + \mu_i^2 \mathbf{T}^2}}. \quad (3)$$

Equations (2) and (3) illustrate the fact that only the product $\mu_i \mathbf{T} = \frac{1}{r_i} \mathbf{T}$ can be estimated, and thus distances and translations can be inferred from camera images only “up-to-scale”.

A. Derivation of the KvD equation

We can use the linearizations

$$\hat{\mathbf{R}} \mathbf{x} \approx \mathbf{x} + \mathbf{a} \times \mathbf{x} \quad (4)$$

$$\hat{\mathbf{R}}^{-1} \mathbf{x} \approx \mathbf{x} - \mathbf{a} \times \mathbf{x} \quad (5)$$

for small rotations described by the rotation vector $\mathbf{a} = \alpha \mathbf{e}_a$ with $\alpha \ll 1$ (α is the rotation angle, \mathbf{e}_a defines the axis of rotation). Using (5), we approximate (2) by

$$\mathbf{e}'_i \approx \frac{\mathbf{e}_i - \mu_i \mathbf{T} - \mathbf{a} \times (\mathbf{e}_i - \mu_i \mathbf{T})}{\sqrt{1 - 2\mu_i \mathbf{e}_i^\top \mathbf{T} + \mu_i^2 \mathbf{T}^2}}. \quad (6)$$

If we further assume $\|\mu_i \mathbf{T}\| = \frac{1}{r_i} \|\mathbf{T}\| \ll 1$, i.e. translation is small compared to object distance r_i , then we have in first

order approximation,

$$\begin{aligned} \mathbf{e}'_i &\approx (\mathbf{e}_i - \mu_i \mathbf{T})(1 + \mu_i \mathbf{e}_i^\top \mathbf{T}) - \mathbf{a} \times \mathbf{e}_i \\ &\approx \mathbf{e}_i - \mu_i \mathbf{T} + \mu_i (\mathbf{e}_i^\top \mathbf{T}) \mathbf{e}_i - \mathbf{a} \times \mathbf{e}_i. \quad (7) \end{aligned}$$

The shift of a feature on the sphere can be described by the “flow vector” $\Delta \mathbf{e}_i$,

$$\begin{aligned} \Delta \mathbf{e}_i &= \mathbf{e}'_i - \mathbf{e}_i \\ &\approx -\mu_i (\mathbf{T} - (\mathbf{e}_i^\top \mathbf{T}) \mathbf{e}_i) - \mathbf{a} \times \mathbf{e}_i. \quad (8) \end{aligned}$$

Equation (8) is the central equation of the KvD algorithm. For given flow vectors $\Delta \mathbf{e}_i$, the KvD algorithm estimates the motion parameters \mathbf{a} , \mathbf{T} and the inverse object distances $\{\mu_i\}$ by minimizing the error function

$$\begin{aligned} E(\mathbf{a}, \mathbf{T}, \{\mu_i\}) &= \sum_i \left(\Delta \mathbf{e}_i - (-\mu_i (\mathbf{T} - (\mathbf{e}_i^\top \mathbf{T}) \mathbf{e}_i) - \mathbf{a} \times \mathbf{e}_i) \right)^2 \quad (9) \end{aligned}$$

under the constraint $\|\mathbf{T}\| = 1$ using an iterative procedure [10].

B. Derivation of the generalized algorithm

For the generalized algorithm, we avoid the approximations that led to (8) and try to directly minimize

$$\sum_i \left(\mathbf{e}'_i - \hat{\mathbf{R}}^{-1} \frac{\mathbf{e}_i - \mu_i \mathbf{T}}{\sqrt{1 - 2\mu_i \mathbf{e}_i^\top \mathbf{T} + \mu_i^2 \mathbf{T}^2}} \right)^2,$$

or equivalently,

$$\sum_i \left(\hat{\mathbf{R}} \mathbf{e}'_i - \frac{\mathbf{e}_i - \mu_i \mathbf{T}}{\sqrt{1 - 2\mu_i \mathbf{e}_i^\top \mathbf{T} + \mu_i^2 \mathbf{T}^2}} \right)^2,$$

with respect to $\hat{\mathbf{R}}$, \mathbf{T} and $\{\mu_i\}$. Since possibly not all feature direction vectors can be estimated with the same accuracy, we introduce weighting factors w_i and obtain the error function

$$\begin{aligned} E(\hat{\mathbf{R}}, \mathbf{T}, \{\mu_i\}) &= \sum_i w_i \left(\hat{\mathbf{R}} \mathbf{e}'_i - \frac{\mathbf{e}_i - \mu_i \mathbf{T}}{\sqrt{1 - 2\mu_i \mathbf{e}_i^\top \mathbf{T} + \mu_i^2 \mathbf{T}^2}} \right)^2. \quad (10) \end{aligned}$$

We rewrite (10) by introducing parameters $\{\gamma_i\}$,

$$\begin{aligned} E(\hat{\mathbf{R}}, \mathbf{T}, \{\mu_i\}, \{\gamma_i\}) &= \sum_i w_i \left(\hat{\mathbf{R}} \mathbf{e}'_i - \gamma_i (\mathbf{e}_i - \mu_i \mathbf{T}) \right)^2, \quad (11) \end{aligned}$$

with the additional constraints $\gamma_i \|\mathbf{e}_i - \mu_i \mathbf{T}\| = 1$. We will also fix $\|\mathbf{T}\| = 1$.

Let us assume for the moment that $\{\mu_i\}$ and $\{\gamma_i\}$ are known. Then the optimal motion parameters $\hat{\mathbf{R}}$ and \mathbf{T} can be calculated directly. Solving $\nabla_{\mathbf{T}} E(\hat{\mathbf{R}}, \mathbf{T}) = \mathbf{0}$ for \mathbf{T} gives

$$\begin{aligned} \mathbf{T}^* &= \frac{\sum_i w_i \gamma_i \mu_i (\gamma_i \mathbf{e}_i - \hat{\mathbf{R}} \mathbf{e}'_i)}{\sum_i w_i \gamma_i^2 \mu_i^2} \\ &= -\hat{\mathbf{R}} \frac{\sum_i w_i \gamma_i \mu_i \mathbf{e}'_i}{\sum_i w_i \gamma_i^2 \mu_i^2} + \frac{\sum_i w_i \gamma_i^2 \mu_i \mathbf{e}_i}{\sum_i w_i \gamma_i^2 \mu_i^2} \quad (12) \end{aligned}$$

$$= -\hat{\mathbf{R}} \bar{\mathbf{e}}' + \bar{\mathbf{e}}, \quad (13)$$

where $\bar{\mathbf{e}}$ and $\bar{\mathbf{e}}'$ are defined as

$$\bar{\mathbf{e}} = \frac{\sum_i w_i \gamma_i^2 \mu_i \mathbf{e}_i}{\sum_i w_i \gamma_i^2 \mu_i^2} \quad \text{and} \quad \bar{\mathbf{e}}' = \frac{\sum_i w_i \gamma_i \mu_i \mathbf{e}_i'}{\sum_i w_i \gamma_i^2 \mu_i^2}. \quad (14)$$

Substituting (13) into (11) gives

$$\begin{aligned} E(\hat{\mathbf{R}}, \{\mu_i\}, \{\gamma_i\}) &= E(\hat{\mathbf{R}}, \mathbf{T}^*, \{\mu_i\}, \{\gamma_i\}) \\ &= \sum_i w_i \left(\hat{\mathbf{R}}(\mathbf{e}_i' - \gamma_i \mu_i \bar{\mathbf{e}}') - \gamma_i (\mathbf{e}_i - \mu_i \bar{\mathbf{e}}) \right)^2 \\ &= \sum_i w_i \left(\hat{\mathbf{R}} \mathbf{c}_i' - \mathbf{c} \right)^2 \\ &= \sum_i w_i \left(\mathbf{c}_i'^{\top} \mathbf{c}_i' + \mathbf{c}_i^{\top} \mathbf{c}_i - 2 \mathbf{c}^{\top} \hat{\mathbf{R}} \mathbf{c}_i' \right)^2, \end{aligned} \quad (15)$$

where \mathbf{c}_i and \mathbf{c}_i' are defined as

$$\begin{aligned} \mathbf{c}_i &= \gamma_i (\mathbf{e}_i - \mu_i \bar{\mathbf{e}}) = \gamma_i \left(\mathbf{e}_i - \mu_i \frac{\sum_k w_k \gamma_k^2 \mu_k \mathbf{e}_k}{\sum_k w_k \gamma_k^2 \mu_k^2} \right), \\ \mathbf{c}_i' &= \mathbf{e}_i' - \gamma_i \mu_i \bar{\mathbf{e}}' = \mathbf{e}_i' - \gamma_i \mu_i \frac{\sum_k w_k \gamma_k \mu_k \mathbf{e}_k'}{\sum_k w_k \gamma_k^2 \mu_k^2}. \end{aligned}$$

The minimum of (15) under the constraint $\hat{\mathbf{R}}^{\top} \hat{\mathbf{R}} = \hat{\mathbf{I}}$ (where $\hat{\mathbf{I}}$ is the identity matrix) can be found using the ‘‘Procrustes approach’’, see [11], [12], [13], [14], i.e.

$$\hat{\mathbf{R}}^* = \hat{\mathbf{P}} \text{diag}(1, 1, \det(\hat{\mathbf{P}} \hat{\mathbf{Q}}^{\top})) \hat{\mathbf{Q}}^{\top}, \quad (16)$$

where $\hat{\mathbf{P}}$ and $\hat{\mathbf{Q}}$ are unitary matrices obtained from Singular Value Decomposition,

$$\left(\hat{\mathbf{P}}, \hat{\mathbf{S}}, \hat{\mathbf{Q}} \right) = \text{svd} \left(\sum_i w_i \mathbf{c}_i \mathbf{c}_i'^{\top} \right). \quad (17)$$

In (16) the multiplication with the diagonal matrix $\text{diag}(1, 1, \det(\hat{\mathbf{P}} \hat{\mathbf{Q}}^{\top}))$ ensures $\det(\hat{\mathbf{R}}^*) = 1$.

Minimizing (11) with respect to μ_k for given $\hat{\mathbf{R}}$ \mathbf{T} and $\{\gamma_i\}$ yields

$$\begin{aligned} \mu_k^* &= \frac{\gamma_k \mathbf{e}_k^{\top} \mathbf{T} - (\hat{\mathbf{R}} \mathbf{e}_k')^{\top} \mathbf{T}}{\gamma_k \|\mathbf{T}\|^2} \\ &\stackrel{\|\mathbf{T}\|=1}{=} \mathbf{e}_k^{\top} \mathbf{T} - \frac{1}{\gamma_k} (\hat{\mathbf{R}} \mathbf{e}_k')^{\top} \mathbf{T}. \end{aligned} \quad (18)$$

From the constraint $\gamma_k^2 (\mathbf{e}_k - \mu_k \mathbf{T})^2 = 1$, we find

$$\begin{aligned} 1 &= \gamma_k^2 (\mathbf{e}_k - \mu_k \mathbf{T})^2 \\ &\stackrel{(18)}{=} \gamma_k^2 \left(\mathbf{e}_k - \left(\mathbf{e}_k^{\top} \mathbf{T} - \frac{1}{\gamma_k} (\hat{\mathbf{R}} \mathbf{e}_k')^{\top} \mathbf{T} \right) \mathbf{T} \right)^2 \\ &= \gamma_k^2 (1 - (\mathbf{e}_k^{\top} \mathbf{T})^2) + \left((\hat{\mathbf{R}} \mathbf{e}_k')^{\top} \mathbf{T} \right)^2, \\ \implies \gamma_k^2 &= \frac{1 - \left((\hat{\mathbf{R}} \mathbf{e}_k')^{\top} \mathbf{T} \right)^2}{1 - (\mathbf{e}_k^{\top} \mathbf{T})^2}, \end{aligned} \quad (19)$$

$$0 < \gamma_k = \sqrt{\frac{1 - \left((\hat{\mathbf{R}} \mathbf{e}_k')^{\top} \mathbf{T} \right)^2}{1 - (\mathbf{e}_k^{\top} \mathbf{T})^2}}. \quad (20)$$

Algorithm: Using the results of the previous section, we propose the following iterative procedure for estimating $\hat{\mathbf{R}}, \mathbf{T}$ and $\{\mu_i\}$ from N tracked features:

- Initialization (t=0):
Set $\mu_i = 1$ and $\gamma_i = 1$, $i = 1, 2, \dots, N$, estimate $\hat{\mathbf{R}}$ and \mathbf{T} using (16) and (12), and compute $E_0 = \sum_i w_i \left(\hat{\mathbf{R}} \mathbf{e}_i' - \mathbf{e}_i + \mathbf{T} \right)^2$
- Loop ($t \leftarrow t + 1$):
1) Normalize \mathbf{T} , i.e. $\mathbf{T} \leftarrow \mathbf{T} / \|\mathbf{T}\|$
2) Calculate γ_i using (20) and μ_i using (18), $i = 1, 2, \dots, N$
3) Estimate $\hat{\mathbf{R}}$ and \mathbf{T} using (16) and (12), respectively
4) Compute $E_t = E(\hat{\mathbf{R}}, \mathbf{T}, \{\mu_i\}, \{\gamma_i\})$ from (11)
- Until $\frac{1}{N} |E_{t-1} - E_t| < \epsilon_{\Delta E}$.

We used $\epsilon_{\Delta E} = 10^{-10}$ and a maximum number of 100 iterations for the experiments described in section IV. In (the rare) case the denominator in (20) is very small, we set $\gamma_i = 1$ and $\mu_i = 0$ (in our implementation we checked for $\mathbf{e}_i^{\top} \mathbf{T} > \cos(0.25^\circ)$, i.e. whether the angle between feature direction vector \mathbf{e}_i and estimated translation \mathbf{T} is smaller than 0.25°).

III. COMPACT OMNIDIRECTIONAL CAMERA

In this section, we describe a novel compact omnidirectional camera that is designed for mobile robots, in particular for small aerial vehicles. In order to obtain ground-truth data by means of an external tracking system the camera of our four-rotor flyer (see Fig. 1) was mounted on a helmet, as shown in Fig 2 a. Results of this experiment will be presented in section IV.

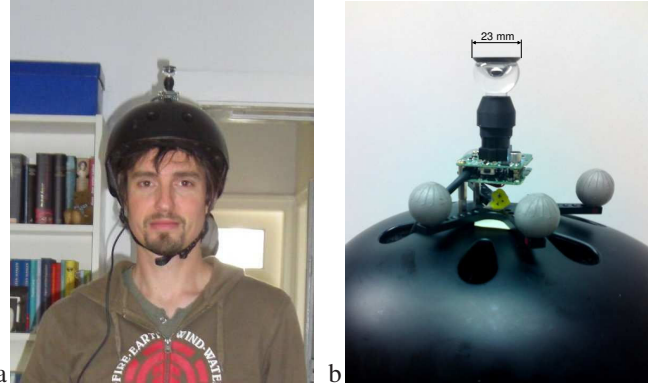


Fig. 2. a) Omnidirectional camera mounted on a helmet used for testing of the generalized algorithm. b) Close-up view with spherical markers of the ART-tracking system attached. The diameter of the spherical body is ≈ 23 mm.

As shown in Fig. 2 b and Fig. 3 a, we improved our light-weight and compact omnidirectional mirror design described in [15] by further reducing weight and size. In contrast to most omnidirectional systems that are usually much larger because they consist of a separate mirror and additional supporting parts [16], the reflective surface (red curve in Fig. 3) was machined into solid acrylic glass and coated with a vapor-deposited aluminum layer. The outer curved

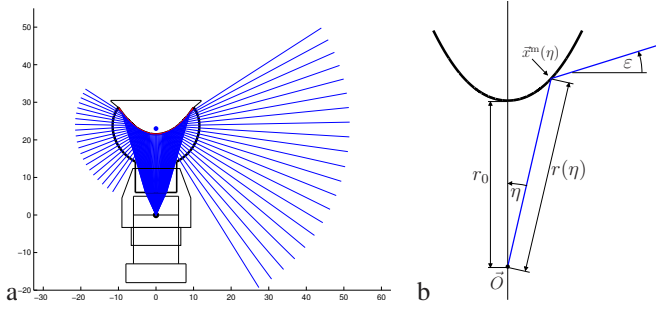


Fig. 3. a) Schematic drawing and raytracing diagram (blue rays) of our omnidirectional imaging system. The red curve highlights the shape of the reflective surface. The blue dot at (0, 23) mm illustrates the approximate view-point. b) Illustration of the parameters defining the shape of a reflective surface with constant angular gain, equation (22).

surface is designed to be orthogonal to the incoming rays to avoid ray deflection due to refraction. A small ‘plastic cap’ was fitted into the concave opening for protection of the thin aluminum layer and to prevent unwanted light from entering the camera from above.

We used a shape with ‘‘constant angular gain’’ [17] for the reflective surface (see red curve in Fig. 3). This leads to a linear mapping of η , the angle between the reflected rays and the optical axis, to elevation angle ε ,

$$\varepsilon = g_\alpha \eta - \frac{\pi}{2}, \quad (21)$$

where g_α is the ‘‘angular gain’’ of the reflective surface. As derived in [17] and illustrated in Fig 3 b, the profile of the reflective surface is defined by

$$\begin{aligned} \mathbf{x}^m(\eta) &= r(\eta) \begin{pmatrix} \sin \eta \\ \cos \eta \end{pmatrix}, \quad \eta \in (0, \eta_{\max}), \quad (22) \\ r(\eta) &= \frac{r_0}{\cos\left(\frac{g_\alpha+1}{2}\eta\right)^{\frac{2}{g_\alpha+1}}}. \end{aligned}$$

The angular gain of the reflective surface in our omnidirectional camera was chosen to be $g_\alpha = 7.5$. The second parameter r_0 that defines the distance to the apex of the mirror is $r_0 = 25$ mm; the maximum of angle η is $\eta_{\max} = 17.3^\circ$. Due to refraction at the planar acrylic glass surface facing the camera lens, the effective view point of the camera is about 3.5 mm further away from the reflective surface and the camera angle is reduced by a factor $1/n_p$, where $n_p \approx 1.5$ is the refractive index of acrylic glass. See [15] for further details.

For capturing images, we used a Philips SPC 900NC USB webcam that is based on a 1/4-inch color CCD sensor (640 × 480 pixels) and well supported by the Linux operating system¹. We removed the camera housing and replaced the original lens with a S-mount lens of 2.5 mm focal length and an IR-cut-off filter attached.

¹<http://www.saillard.org/linux/pwc/>

IV. RESULTS

A. Simulation

For comparing the presented generalized algorithm and the original KvD approach and for analyzing the noise dependence of the ego-motion estimation results, we simulated the movement of a spherical camera in an environment with $N = 14$ features. As shown in Fig. 4 a, the simulated camera starts surrounded by the points with varying nearness values μ_i . It moves in 100 steps along a half circle while continuously changing orientation from 0° to 180° . For each time step t , feature points were projected onto the viewing sphere and optionally Gaussian noise was added, resulting in the feature directions $\mathbf{e}_i(t)$, $i = 1, 2, \dots, N$. The movement of the features on the viewing sphere is shown in Fig. 4 b.

Fig. 4 c,d shows results for zero noise. Please note that in order to test its performance for large motion between camera frames, the ego-motion estimation starts ‘‘from scratch’’ at every time step, i.e. no information from previous steps is used.² While the generalized algorithm estimates correctly orientation and direction of translation for all positions, the original KvD approach produces valid results only as long as the change in orientation is smaller than about 10° . For noise level of 1° , the translation direction cannot be estimated correctly for small translations of the camera as depicted in Fig. 4 f for $t \lesssim 5$. This is not unexpected since for small translations shifts of the features in the image caused by translation are small and hard to detect in the presence of noise. Using more points will improve the ego-motion estimation accuracy due to the implicit averaging of the algorithm.

As illustrated in Fig. 5, the simultaneously estimated inverse distances μ_i allow to reconstruct the 3D-positions of the feature points. Since each inverse distance contributes just to a single term to the sum defining the error function (10), the accuracy of the estimated μ_i will mainly depend on the accuracy of the vectors \mathbf{e}_i and $\mathbf{e}'_i(t)$. However, similar to the original KvD approach [10], slight improvements can still be achieved by increasing the number of points since estimation of rotation and translation direction will be enhanced.

B. Real-world experiment

As it was more convenient to obtain ground-truth data indoors, the omnidirectional camera of the four-rotor flyer (see Fig. 1) was mounted on top of a black helmet, enabling us to capture images while walking freely in a room, see Fig. 2 a. Four spherical markers were attached below the camera (Fig. 2 b) for estimating all 6 DOF with an ART tracking system³. Images were captured at a constant rate of 10 fps and stored on a laptop carried in a backpack. The trajectory measured by the ART tracking system is shown in Fig. 8 a.

In order to obtain feature correspondences, features were tracked for 148 frames (corresponding to about 15 sec) using

²Of course, for a continuously operating ego-motion estimation, the solution from the previous step can be used for initialization. In addition, reliably estimated μ_i ’s could be fixed and the normalization of \mathbf{T}^{est} omitted.

³<http://www.ar-tracking.de/>

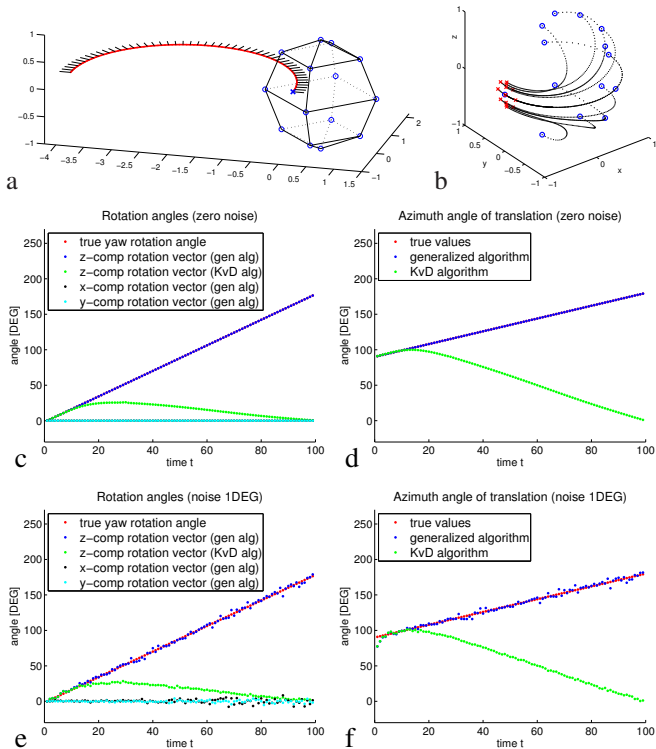


Fig. 4. Testing the original KvD approach and the generalized algorithm using simulated features positions and their projections onto the unit-sphere. a) Overview of the position of the feature points (located at the 14 corners of the polyhedron), the camera trajectory (red curve, the blue cross marks the start position), and the camera orientation (illustrated as short lines orthogonal to the trajectory). b) Motion of the projected feature points on the unit-sphere. Start points are highlighted by blue circles, end points are highlighted by red crosses. c) – f) Comparison of the KvD algorithm (green dots) and the proposed generalized algorithm (blue dots). Ground truth values are shown in red. c) Estimation of rotation for zero noise. Plotted are the components of the rotation vector. d) Estimation of the azimuth angle of the translation vector for zero noise. The ground-truth values are indistinguishable from the values estimated by the generalized algorithm. The original KvD algorithm produces correct estimates only up to a rotation angle of about 10° . e), f) as c), d) but with Gaussian noise of 1° standard deviation added to the feature vectors \mathbf{e}_i and \mathbf{e}'_i .

Birchfeld’s C-implementation [18] of the Kanade-Lucas-Tomasi (KLT) feature tracker [19], [20]. The omnidirectional camera was calibrated using the “OCamCalib” Matlab toolbox [21] enabling us to calculate feature directions \mathbf{e}_i and \mathbf{e}'_i from points in the images. We kept 50 features that showed the smallest standard deviation in μ_i relative to the mean over the sequence (see green circles Fig. 6 a,b). The reference frames of the omnidirectional camera and the tracking system were aligned using the estimated and measured (normalized) translation vectors for frames 20–50.

Results of the ego-motion estimation (rotation vector and translation direction) are shown in Fig. 7. Similar to the results of the simulation (Fig. 4f for $t \lesssim 5$), the estimated translation direction shows comparatively large errors for small translations because feature shift is small and noise has strong impact (dashed blue curve in Fig. 7 b for $t \lesssim 20$). For larger frame numbers $t \gtrsim 70$, errors are most probably

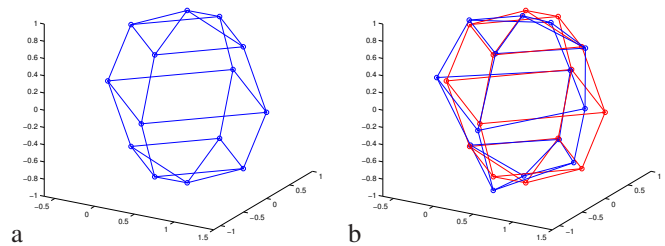


Fig. 5. Reconstruction of scene structure $\mathbf{x}_i^{\text{est}}(t) = \mu_i^{-1}(t)\mathbf{e}_i(t)$ at $t = 50$. For comparison with true 3D positions, $\mathbf{x}_i^{\text{est}}$ were multiplied by $\|\mathbf{T}^{\text{true}}(t)\|$. a) For zero noise, estimated 3D points (blue) are indistinguishable from true feature positions. b) Results with Gaussian noise of 1° standard deviation (true feature positions are highlighted by red dots).

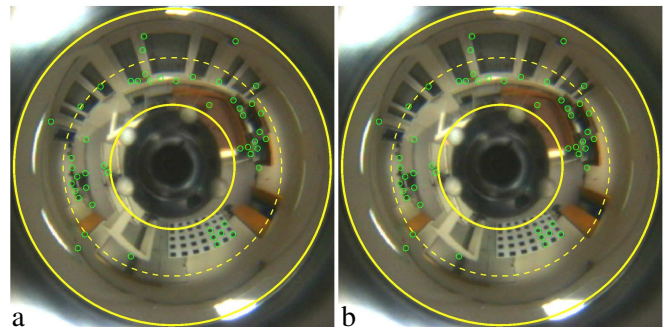


Fig. 6. First (a) and last (b) camera image of a image sequence where 50 features were tracked for 148 frames with a KLT tracker. Features are highlighted by small green circles. The dashed yellow circle highlights the horizon (elevation angle $\varepsilon = 0^\circ$) of the omnidirectional camera, the two continuous yellow circles demarcate the FOV used in the experiment (inner circle $\varepsilon = -40^\circ$, outer circle $\varepsilon = +36^\circ$). Only features inside this area were tracked. The spherical markers of the ART tracking system are visible in the image center close to the inner yellow circle.

due to drifts in the feature tracking process. Estimation of the z -component of the rotation vector appears to be less sensitive to this error (dashed blue curve in Fig. 7 a).

The algorithm was again initialized “from scratch” in every time step in order to test its performance for large motion between the two camera frames. Thus, it can estimate translation and inverse distances only up to scale. However, it is possible to introduce an approximately fixed scale over the whole sequence by assuming that the mean over the inverse distances $\bar{\mu}(t) = \frac{1}{N} \sum_i \mu_i(t)$ is proportional to translations. The rationale behind this is that for perfect feature tracking and because the translation vector is normalized in every time step, $\mu_i(t)$ and thus also $\sum \mu_i(t)$ are proportional to translation, i.e. $\sum \mu_i(t) \propto T(t)\mu_i(1)$. Thus, by plotting $\bar{\mu}(t)\mathbf{T}^{\text{est}}(t)$ we can obtain an estimate for the whole trajectory up to scale. For better comparison with the true trajectory, we multiplied the estimated trajectory by a scaling factor k in Fig. 8 a, where k was chosen to minimize $\sum_{t=20}^{50} \|\mathbf{T}^{\text{true}}(t) - k\bar{\mu}(t)\mathbf{T}^{\text{est}}(t)\|^2$ – the deviation of the 3D trajectories between the 20th and 50th frame.

V. DISCUSSION AND OUTLOOK

We have presented an algorithm for visual ego-motion estimation with monocular cameras that does not require

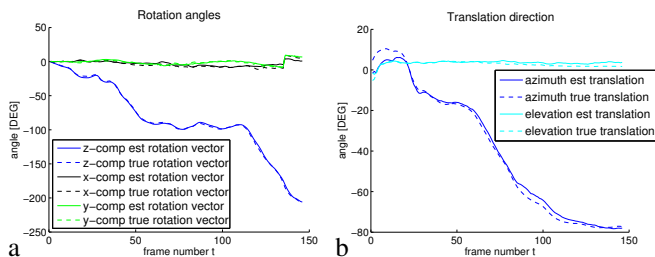


Fig. 7. Real-world test of the generalized algorithm. Dashed curves show ground-truth data from the ART-tracking system. a) Estimation of rotation. Plotted are the components of the rotation vector. b) Estimation of azimuth and elevation angle of the translation vector.

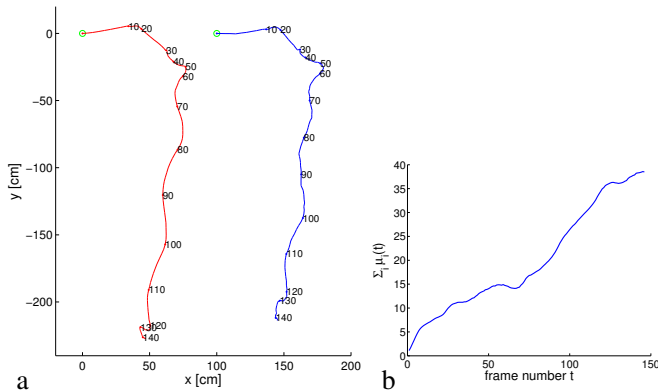


Fig. 8. a) Trajectory measured by the ART-tracking system (red) and estimated trajectory (blue, shifted by 100 cm in x -direction) using $k\bar{\mu}(t)\mathbf{T}^{\text{est}}(t)$ as an estimate of the position at time t . The scaling factor k was calculated to minimize the sum of squared differences between the estimated and the measured translation vectors between the 20th and 50th frame (see text). The numbers close to the trajectories show the corresponding frame numbers. b) Sum of nearnesses $\sum_i \mu_i(t) = N\bar{\mu}(t)$ over the whole sequence.

any initialization sequence in opposite to Davison’s original approach [22]⁴ or the VGPS system [13]. The presented algorithm does not rely on the linearization introduced in the original KvD algorithm and, therefore, allows a convergence over a large range of rotations and translations as validated in the results section. This system was tested on an omnidirectional camera that enhances the visibility of the tracked points over a larger range of motion. As already mentioned earlier, we calculate the localization result relative to an initial reference image instead of integrating the incremental changes over an image sequence. This prevents accumulation the ever present error due to quantization and detection accuracy.

While the algorithm computes ego-motion reliably even in the presence of noise, feature drift during tracking can significantly degrade estimation results. Although changes in feature positions are usually small between consecutive frames and thus feature tracking is unlikely to be strongly affected by the varying resolution in azimuth angle of our omnidirectional camera, adapting the KLT algorithm to this

⁴A recent extension of Davison’s approach by Civiera et al. [23] based on inverse distances makes a separate initialization procedure unnecessary.

type of sensor might improve tracking performance. Of course, instead of continuously tracking features, approaches that directly match features, like SIFT [24] or SURF [25], could be used. However they are computationally more expensive.

The weight factors w_i in (17) and (12) allow to modify the contribution of a single feature to the result. The tracking quality of features depend on the structure of the local environment. Some features cannot be tracked reliably. This error can be found through re-projection of the position of each feature using the current localization result. Features with large errors should have a lower weight in the algorithm as proposed in [26]. Even in case of ideal tracking, features detected in images have different sensitivity to motion in 3D space. The Image Jacobian matrix relates changes in the images to changes in the position. The observability of the features was studied in [27] where the condition number (the largest ratio of the eigenvalues) of the Image Jacobian was analyzed. As a result, features with reasonable projective height in the image were optimal for the motion detection. For planar motion, neither features along the horizon nor features above the robot are good candidates to be included in the feature set if the number of features needs to be kept low. Therefore, close features with large “nearness” value μ_i are preferred in the calculation of motion and should be given a higher weight. This will be generalized and analyzed in the future work.

In future work, the proposed generalized KvD algorithm will be compared to existing state-of-the art approaches and tested on the four-rotor flyer shown in Fig. 1. The algorithm will be used in a global navigation system robustified through fusion with inertial measurements. It is not feasible to keep just one reference image in global navigation frameworks. Therefore, we will add a hand-off process to switch between different reference points while traveling over large distances. This approach has been applied in a navigation system described in [26].

VI. ACKNOWLEDGMENTS

We thank Simon Strübbe for providing an implementation of the KvD-algorithm, Claudius Strub for MATLAB programming, and the machine shops at the Institute of Robotics and Mechatronics (DLR) and Bielefeld University for machining the parts of the omnidirectional camera.

REFERENCES

- [1] H. C. Longuet-Higgins, “A Computer Algorithm for Reconstructing a Scene from Two Projections,” *Nature*, vol. 293, pp. 133–135, 1981.
- [2] R. I. Hartley, “In defense of the eight-point algorithm,” *Pattern Analysis and Machine Intelligence*, vol. 19, pp. 580–593, 1997.
- [3] D. Nister, “An efficient solution to the five-point relative pose problem,” *Pattern Analysis and Machine Intelligence*, vol. 26, pp. 756–770, 2004.
- [4] O. Faugeras and F. Lustman, “Motion and structure from motion in a piecewise planar environment,” *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 2, no. 3, pp. 485–508, 1988.
- [5] Z. Zhang and A. R. Hanson, “3d reconstruction based on homography mapping,” in *ARPA*, 1996, pp. 1007–1012.
- [6] R. F. Vassallo, J. Santos-Victor, and H. J. Schneebeli, “A general approach for egomotion estimation with omnidirectional images,” in *OMNIVIS*, 2002, pp. 97–103.

- [7] J.-P. Tardif, Y. Pavlidis, and K. Daniilidis, "Monocular visual odometry in urban environments using an omnidirectional camera," in *IROS*, 2008, pp. 2531–2538.
- [8] A. Torii, M. Havlena, and T. Pajdla, "Omnidirectional image stabilization by computing camera trajectory," in *PSIVT*, 2009, pp. 71–82.
- [9] D. Scaramuzza, F. Fraundorfer, and R. Siegwart, "Real-time monocular visual odometry for on-road vehicles with 1-point ransac," in *ICRA*, 2009, pp. 488–494.
- [10] J. J. Koenderink and A. J. Doorn, "Facts on optic flow," *Biological Cybernetics*, vol. 56, no. 4, pp. 247–254, 1987.
- [11] P. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, pp. 1–10, 1966.
- [12] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, 1991.
- [13] D. Burschka and G. D. Hager, "V-GPS – Image-Based Control for 3D Guidance Systems," in *IROS*, 2003, pp. 1789–1795.
- [14] L. Dorst, "First order error propagation of the procrustes method for 3d attitude estimation," *Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 221–229, 2005.
- [15] W. Stürzl, M. Suppa, and D. Burschka, "Light-weight panoramic mirror design for visual navigation," in *Omnidirectional Robot Vision*, 2008, pp. 218–229.
- [16] R. Benosman and S. B. Kang, *Panoramic Vision: Sensors, Theory and Applications*, 1st ed. Springer, Berlin, 2001.
- [17] J. Chahl and M. Srinivasan, "Reflective surfaces for panoramic imaging," *Applied Optics*, vol. 36, pp. 8275–8285, 1997.
- [18] S. Birchfield, "KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker," www.ces.clemson.edu/~stb/klt/.
- [19] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, 1981, pp. 674–679.
- [20] J. Shi and C. Tomasi, "Good features to track," in *CVPR*, 1994, pp. 593–600.
- [21] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *IROS*, 2006, pp. 5695–5701.
- [22] A. J. Davison, "Real-Time Simultaneous Localisation and Mapping with a Single Camera," in *ICCV*, 2003, pp. 1403–1412.
- [23] J. Civera, A. J. Davison, and J. M. M. Montiel, "Inverse depth parametrization for monocular slam," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.
- [24] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999, pp. 1150–1157.
- [25] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [26] E. Mair, K. Strobl, M. Suppa, and D. Burschka, "Efficient Camera-Based Pose Estimation for Real-Time Applications," in *IROS*, 2009, pp. 2696–2703.
- [27] D. Burschka, J. Geiman, and G. D. Hager, "Optimal Landmark Configuration for Vision-Based Control of Mobile Robots," in *ICRA*, 2003, pp. 3917–3922.