Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Autonomous Systems Lab
Prof. Roland Siegwart

Master Thesis

Use of a DVS for High Speed Applications

Autumn Term 2018

Supervised by:
Florian Tschopp
Ignacio Alzugaray

Author:
Weizhen Huang

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Contents

Preface

Bla bla …

Abstract

Hier kommt der Abstact hin . . .

Symbols

Symbols

ϕ, θ, ψ	roll, pitch and yaw angle
b	gyroscope bias
Ω_m	3-axis gyroscope measurement

Indices

x	x axis
y	y axis

Acronyms and Abbreviations

ETH	Eidgenössische Technische Hochschule
EKF	Extended Kalman Filter
IMU	Inertial Measurement Unit
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter

Chapter 1

Introduction

Hier kommt die Einleitung DVS is

Chapter 2

6DoF Pose Tracking in Planar Scenes

Throughout this work the following notation is employed: W denotes the world frame, C_1 or C_2 denotes a camera frame. T_{AB} is the transformation from frame A to frame B, measured in frame A.
$\boldsymbol{X}$ the position of the event with respect to world or camera frame, $\boldsymbol{x}$ the calibrated coordinates of the event.

2.1 From Events to Frame

We group a set of events $\mathscr{E} \doteq \{e_k\}_{k=1}^{N}$ into a temporal window, optimize the motion and scene parameters within this window, then shift the window to the next set of events and repeat this process. The temporal window size is defined by the event numbers N, which should be chosen small enough so that a constant velocity model could be applied within this window. We choose event numbers against a fixed time interval to define the window size, because this corresponds to the data-driven nature of an event-based camera: the more rapid the apparent motion of the scene is, the larger the event rate will be. If the scene stops moving, no events will be generated, the pose will also not be further updated.
An event frame is thus formed by summing up events within this window. If we simply sum along the time axis, the intensity at each pixel will be the sum of the polarities of all the events that are triggered at this pixel location within the window

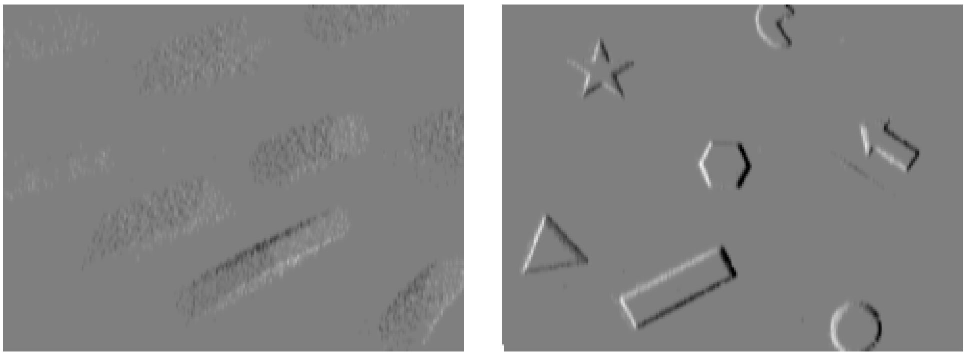
$$\mathcal{I}(\boldsymbol{x}) = \sum_{k=1}^{N} \pm_k \delta(\boldsymbol{x} - \boldsymbol{x}_k), \tag{2.1}$$

with $\pm_k$ and $\boldsymbol{x}_k$ denoting the polarity and pixel coordinates of the kth event, respectively. After warping the events with $\boldsymbol{x}'_k = \boldsymbol{W}(\boldsymbol{x}_k, t; \theta)$, we substitute $\boldsymbol{x}_k$ in the above equation to $\boldsymbol{x}'_k$.

2.1.1 Measuring the Sharpness of an Image

There are several metrics one could choose from to measure the contrast of an image. A local contrast metric could be, for example, convoluting the image with a high pass filter

$$\mathcal{C}_H = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}, \tag{2.2}$$



(a) An optimized frame using sharpening filter metric

(b) An optimized frame using variance metric

Figure 2.1: Comparison between two different metrics

and sum the pixel value of the filtered image. However, this metric only compare a pixel with its 8 neighbors, thus an image with scattered events (large noise) is also a valid configuration, as shown in fig. 2.1(a). The Michelson contrast [1], defined as $\mathcal{C}_M = (\mathcal{I}_{\max} - \mathcal{I}_{\min}) / (\mathcal{I}_{\max} + \mathcal{I}_{\min})$, only considers the highest and lowest luminance in the image and is thus more suitable to quantify contrast for periodic functions.
We choose to measure the contrast by the variance of the image, defined by

$$\operatorname{Var}(\mathcal{I}(\boldsymbol{x};\boldsymbol{\theta})) \doteq \frac{1}{\mid \Omega \mid} \int_{\Omega} (\mathcal{I}(\boldsymbol{x};\boldsymbol{\theta}) - \mu(\mathcal{I}(\boldsymbol{x};\boldsymbol{\theta})))^2 \, d\boldsymbol{x}, \tag{2.3}$$

where $\mu(\mathcal{I}(\boldsymbol{x};\boldsymbol{\theta}))$ is the average image intensity. Since our goal is to align events triggered from the same visual stimuli, the variance would be a very suitable metric, as a squared metric favors the configuration that projects as many events as possible to the same pixel.
Intergral sum pixels

2.1.2 Planar Homography

The warp function $\boldsymbol{x}' = \boldsymbol{W}(\boldsymbol{x}, t; \boldsymbol{\theta})$ does not only depend on the motion parameters, but also the scene parameters, which is the unknown depth. In the case of a planar scene the problems simplifies, since a plane $\mathbf{P}$ can be parameterized by two sets of parameters: $\boldsymbol{n} \in \mathbb{S}^2$ the unit surface normal of $\mathbf{P}$ with respect to the current camera frame, and d the distance from the camera center to $\mathbf{P}$. The warp function then becomes

$$\boldsymbol{X}' = \boldsymbol{R}(t)\boldsymbol{X} + \boldsymbol{T}(t) \tag{2.4}$$

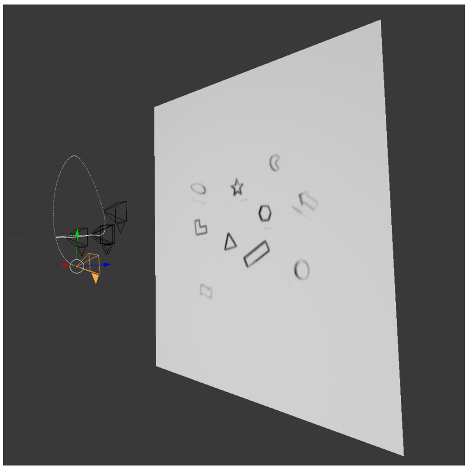
$$\boldsymbol{X} = \boldsymbol{R}(t)^\top (\boldsymbol{X}' - \boldsymbol{T}(t)) \tag{2.5}$$

$$\boldsymbol{X} = \boldsymbol{R}(t)^\top \left(\boldsymbol{I} + \boldsymbol{T}(t)\boldsymbol{n}^\top / d\right) \boldsymbol{X}', \tag{2.6}$$

thus $\boldsymbol{x}' \sim \left(\boldsymbol{R}(t)^\top \left(\boldsymbol{I} + \boldsymbol{T}(t)\boldsymbol{n}^\top / d\right)\right)^{-1} \boldsymbol{x}$. Here $(\boldsymbol{R}(t), \boldsymbol{T}(t)) \in SE(3)$ denotes the relative pose between two cameras at which the current event being warped and the first event within the window happened, and t is the relative timestamp with respect to the first event. Under a constant velocity model with linear velocity $\boldsymbol{v} \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$, the translation is given by

$$\boldsymbol{T}(t) = \boldsymbol{v}t, \tag{2.7}$$

(a) The texture of a map



(b) A map in the global frame

Figure 2.2: Map

the rotation matrix is given by the *exponential map* exp: $\mathfrak{so}(3) \to SO(3)$:

$$\boldsymbol{R}(t) = \exp(\boldsymbol{\omega}^{\wedge} t), \tag{2.8}$$

where $^{\wedge}$ is the *hat* operator

$$\boldsymbol{\omega}^{\wedge} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \in \mathfrak{so}(3). \tag{2.9}$$

2.2 From Frames to Map

The contrast maximization procedure in the above section optimizes the relative pose between successive frames. We show in this section that the same idea can be applied to perform global pose tracking in planar scenes. We first explain how the map is defined, and how to track a known map, then we shown how this map is built by selecting a set of keyframes.

2.2.1 Map

A map is a plane with three components: the normal direction $\boldsymbol{n}_w$, the distance d_w to the origin, and the texture; the texture of a map represents all the edges on the plane. Figure 2.2 shows the an example of such map. Figure 2.2(a) also shows the set of keyframes used to construct the map. We will talk more about keyframes in section 2.2.3. The global coordinate is chosen as the camera coordinate of the first frame.

2.2.2 Tracking

Suppose a map is present, then the normal direction $\boldsymbol{n}_w$ of the plane and the distance d_w to the origin are known. Also the pose of the current frame $(\boldsymbol{R}_{wc}, \boldsymbol{T}_{wc}) \in SE(3)$ is determined by the motion estimation from the last frame (a quick note to the terminology we are using: whenever we say the *pose* of a frame, we always refer to the camera *pose* at which the first event within the frame happens). The parameters left to be estimated for each frame is $\phi = (\boldsymbol{\omega}, \boldsymbol{v}) \in \mathbb{R}^6$. By substituting $\boldsymbol{n}$ with

$\boldsymbol{n}_c = \boldsymbol{R}_{cw}\boldsymbol{n}_w$, and d with $d_c = d_w + \boldsymbol{T}_{wc} \cdot \boldsymbol{n}_w$ in eq. (2.6), we get the homography matrix within each frame as

$$\boldsymbol{H}_1 = \boldsymbol{R}^\top \left(\boldsymbol{I} + \boldsymbol{T}\boldsymbol{n}_c^\top / d_c\right) \tag{2.10}$$

and $\boldsymbol{x}'_c \sim \boldsymbol{H}_1^{-1} x_c$.
A nonlinear optimizing problem naturally suffers from local optima. Without a good initialization, the motion computed with the method in section 2.1 could sometimes be a local optimum delivering an image that appears sharp, despite being wrongly estimated (see fig. 2.3). In order to make sure that the estimated motion from the per frame contrast maximization also conform to the global map. Thus we perform another optimization, where we project the events of the current frame to the global map. The parameter set is still $\left(\boldsymbol{\omega}^\top, \boldsymbol{v}^\top\right)$, and we use the output from last procedure as an initial guess.
The procedure described in the first paragraph of this section can be understood as projecting the events on a *blank canvas*. Similarly, in the projecting-to-map procedure we project the events on the *texture* of the map, and measure the strength of the synthesized image with the same variance function as in eq. (2.3), thus finding the set of the parameters that best align the events in the current frame to their correspondences in the texture.
The projection from an event to the map is

$$\boldsymbol{x}_w \sim \boldsymbol{R}_n \boldsymbol{H}_2^{-1} \boldsymbol{H}_1^{-1} \boldsymbol{x}_c, \tag{2.11}$$

with $\boldsymbol{R}_n$ the transformation from the orientation of the global frame to the orientation of the map, computed by

$$\boldsymbol{K} = (\boldsymbol{n}_w \times \boldsymbol{z})^\wedge \tag{2.12}$$
$$\boldsymbol{R}_n = \boldsymbol{I} + \boldsymbol{K} + \boldsymbol{K}^2/(1 + \boldsymbol{n}_{\boldsymbol{w}} \cdot \boldsymbol{z}), \tag{2.13}$$

where $\boldsymbol{z} = (0, 0, -1)$ denotes the plane fronto-parallel to the camera, and

$$\boldsymbol{H}_2 = \boldsymbol{R}_{cw} \left(\boldsymbol{I} + \boldsymbol{T}_{wc}\boldsymbol{n}^\top / d_w\right) \tag{2.14}$$

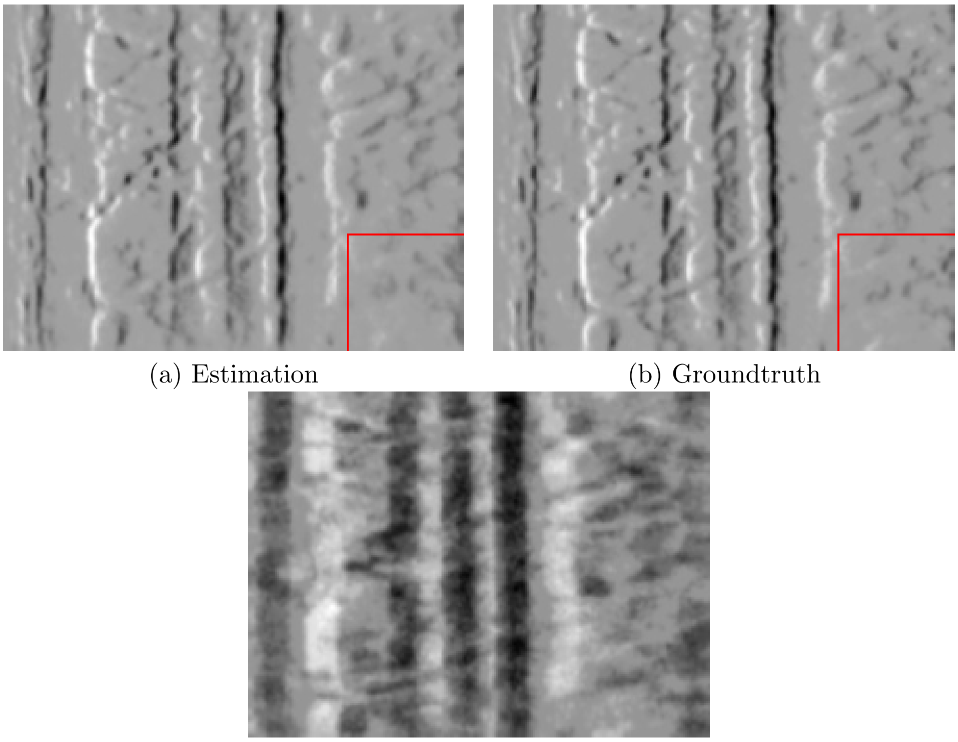
is the projection from the current frame to the global frame, with $\boldsymbol{R}_{wc}, \boldsymbol{T}_{wc}$ being the pose of the current frame. $\boldsymbol{H}_1$ is the planar homography for each frame as in eq. (2.10). But the $\boldsymbol{R}(t)$ and $\boldsymbol{T}(t)$ might be different since we are refining these parameters.
After projecting, we obtain an image composed of the map texture and the events of the current frame. We maximize the contrast of this image using the cost function defined in eq. (2.3). The optimized velocity is used for propagating the pose to the next incoming frame via

$$\boldsymbol{T}_{wc_2} = \boldsymbol{R}_{wc_1}\boldsymbol{v}\Delta t + \boldsymbol{T}_{wc_1} \tag{2.15}$$
$$\boldsymbol{R}_{wc_2} = \boldsymbol{R}_{wc_1}\exp(\boldsymbol{\omega}^\wedge \Delta t), \tag{2.16}$$

where c_1 and c_2 denotes the current frame and the next frame, respectively, and Δt is the temporal size of the current frame.

2.2.3 Mapping

After having collected the first N events, we start the mapping process. For the first frame we estimate the full parameter set $\phi = \left(\boldsymbol{\omega}^\top, \boldsymbol{v}^\top / d_w, \varphi, \psi\right)^\top \in \mathbb{R}^8$, where (φ, ψ) parametrize the unit vector of the plane normal, and $\boldsymbol{v}^\top / d_w$ account for the scale ambiguity problem introduced in linear velocity estimation from monocular



(a) Estimation (b) Groundtruth

(c) Without motion compensation

Figure 2.3: An example of local optima. This is the dataset *slider_hdr_close* with a window size of 50000 events. (a) shows the optimized image with *linear velocity* $(0.231, 0.109, 0.256)$, *angular velocity* $(0.405, -0.130, -0.278)$ and *plane normal* $(-0.579, 0.282, -0.765)$. (b) shows the result using groundtruth parameters with *linear velocity* $(0.163, 0, 0)$, *angular velocity* $(0, 0, 0)$ and *plane normal* $(0, 0, -1)$. Both images appear mostly identical, though at the lower right corner, for example, one can still recognize the difference. Also both images look much sharper than the image without motion compensation in (c). It is worth mentioning that the contrast of the estimation is actually slightly larger than that of the groundtruth

camera. We can for example determine the scale of the scene by setting d_w to $1m$, then scales of the consecutive frames are also determined by $d_c = d_w + \boldsymbol{T}_{wc} \cdot \boldsymbol{n}_w$.
From the optimized first frame we initialize the map, by projecting the frame to the planar scene via a rotation $\boldsymbol{R}_n$ as in eq. (2.13) (you might have implemented this part wrong). Then we can track the next frames with this map using the method described in section 2.2.2.
As the camera moves, there might be new information available in the scene, so that the map needs to be expanded. After optimization for each single frame, we also measure the *per-pixel-overlap* between the frame and the map, that is, how many percent pixels in the frame can be explained by the map. The overlap might be small, when the camera is just exploring a new area so that only part of the frame and the map is overlapping; another possibility is that the map is not accurate enough to explain the current frame, which often happens when only one frame is used until now to construct the map. When the overlap reaches a certain threshold (0.8 for example), we try to insert a new *keyframe*.
When the system decides that a new keyframe is needed, we collect all the keyframes until now, plus the current frame, which is a keyframe candidate, and optimize the poses and velocities of these frames, as well as the map together. Suppose there are k frames (including the keyframe candidate), the to be optimized parameter set is

$$\phi = \left(\boldsymbol{R}_{(1\sim k)}, \boldsymbol{T}_{(1\sim k)}, \boldsymbol{\omega}_{(0\sim k)}, \boldsymbol{v}_{(0\sim k)}/d_w, \varphi, \psi\right) \in \mathbb{R}^{12k-4}.$$

Here $1 \sim k$ means from frame 1 to frame k, and we skip the pose optimization of frame 0, which is the first frame, since it defines the global coordinate. We project all the events of these k frames to the plane parametrize by (φ, ψ) with eq. (2.11), and again optimize the contrast of the synthesized image the obtain the optimal parameter set.
Note that in the mapping process we drop the polarity of the event, since different keyframes might include events from the same visual stimuli but triggered when moving in different directions, when summed together the polarities might cancel each other out. For the same reason, we also don't use the polarity when matching a frame to the map.
After this step, we also measure the *per-pixel-overlap* between the keyframe candidate and the image synthesized by the other keyframes, with the newly optimized parameter set. If this is a good match (define a good match), we continue the tracking. Otherwise, we consider the pose estimation of the current frame to be already off. In this case we preserve the former map, and use this map to start the *relocalization* procedure.

2.2.4 Relocalization

2.3 Experiments

Despite of the rapid movements, in the above datasets the camera actually only moves in a relative small region in front of the textured plane, as depicted in fig. 2.4. However, this method also applies when camera travels a longer distance with respect to the scene depth, as shown in fig. 2.5. Also, although this method is designed for planar scenes, in a complex scene with rotation only movement, where no scene parameters are needed, this method can still be applied to build a global map, delivering a result similar to *image stitching*. Figure 2.6 is such an example.

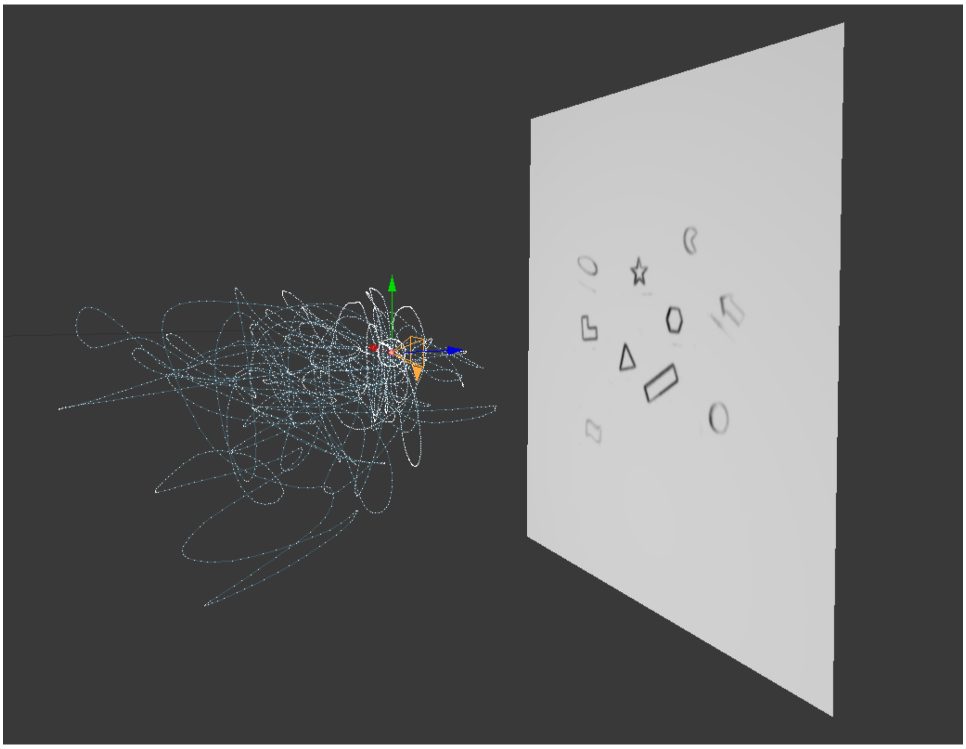


Figure 2.4: The motion path of the dataset *shapes_6dof*. Despite a trajectory length of about $50m$, in the whole $60s$ the camera actually stays in a relative small region compared to the scene depth, and there is no observable increase of drift using the method in this work. ref to error estimation

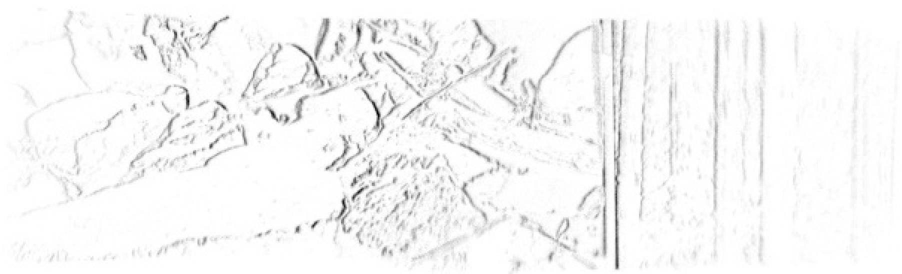


Figure 2.5: The dataset *slider_hdr_far*, with a scene depth of $0.584m$, and a camera movement in the positive x direction only. Note that this is a much wider map than what we have seen before. The figure shows the result at $4.8sec$ within the whole range of $6.3sec$; afterwards the algorithm is lost in the forest, where almost all the textures are vertical, causing severe local optima problem. If we constrain the motion estimation to translation only, it delivers a much better result.

Figure 2.6: The dataset *boxes_rotation*. The motion in this dataset is rotation-dominated

Chapter 3

A Discussion to 6DoF Motion Estimation in General 3D Scenes

3.0.1 note

tried initialize with multiple frames, didn't work very well. similarly sliding window didn't work; too many events didn't work

Chapter 4

Discussion

One possible reason why the algorithm sometimes get lost is that there are too few textures available, as shown in fig. 4.1(a). At this time we usually need to rely on the relocalization pipeline. The current implementation of relocalization needs a good initial guess, so it does not alway work reliably.

4.1 Erstellen einer Tabelle

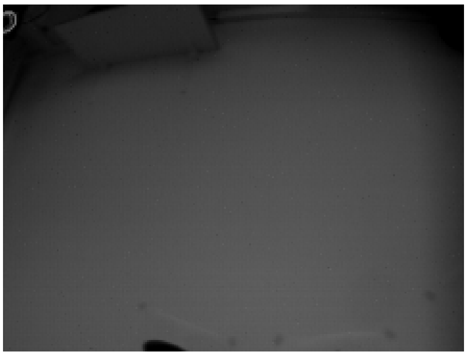
Ein Beispiel einer Tabelle:

Table 4.1: Daten der Fahrzyklen ECE, EUDC, NEFZ.

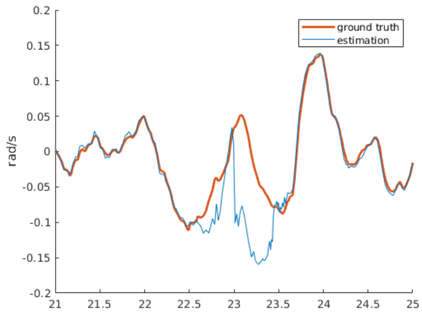
Kennzahl	Einheit	ECE	EUDC	NEFZ
Dauer	s	780	400	1180
Distanz	km	4.052	6.955	11.007
Durchschnittsgeschwindigkeit	km/h	18.7	62.6	33.6
Leerlaufanteil	%	36	10	27

Die Tabelle wurde erzeugt mit:

```
\begin{table}[h]
  \begin{center}
    \caption{Daten der Fahrzyklen ECE, EUDC, NEFZ.}\vspace{1ex}
    \label{tab:tabnefz}
    \begin{tabular}{ll|ccc}
      \hline
      Kennzahl & Einheit & ECE & EUDC & NEFZ \\ \hline \hline
      Dauer & s & 780 & 400 & 1180 \\
      Distanz & km & 4.052 & 6.955 & 11.007 \\
      Durchschnittsgeschwindigkeit & km/h & 18.7 &  62.6 & 33.6 \\
      Leerlaufanteil & \% & 36 & 10 & 27 \\
      \hline
    \end{tabular}
  \end{center}
\end{table}
```



(a) There is little texture visible



(b) Relocalized after get lost (roll component)

(c) Polluted map after tracking is lost

Figure 4.1: When tracking the dataset *shapes_translation*, we always get lost at around 23*sec*, where only a small part of the oval is visible (a). When later there are more textures available, the camera has already moved some distance so that the current estimation of the pose might already be far off. The *bundle adjustment* When the algorithm confirms that there is little overlap between the current frame and the map at this pose. it tries to insert a keyframe and track from the new keyframe. The current implementation doesn't reinitialize the map after the tracking is lost. The advantage is that there is still chance to come back to the original map after tracking a wrong "local map" (b); the disadvantage is that the global map will most likely be polluted by the "local map" (c)

4.2 Weitere nützliche Befehle

Hervorhebungen im Text sehen so aus: *hervorgehoben*. Erzeugt werden sie mit dem `\epmh{.}` Befehl.
Einheiten werden mit den Befehlen `\unit[1]{m}` (z.B. 1 m) und `\unitfrac[1]{m}{s}` (z.B. 1 m/s) gesetzt.

Bibliography

[1] A. A. Michelson, *Studies in optics.* Courier Corporation, 1995.

Appendix A

Derivative of the Contrast Metric

Since we use bilinear voting to evaluate the *dirac delta*, the derivatives of the cost function eq. (2.3) can be analytically computed as

Let

$$\rho(\boldsymbol{x};\boldsymbol{\theta}) = \mathcal{I}(\boldsymbol{x};\boldsymbol{\theta}) - \mu(\mathcal{I}(\boldsymbol{x};\boldsymbol{\theta})) \tag{A.1}$$

Then

$$\frac{\partial}{\partial\boldsymbol{\theta}}\mathrm{Var}(\mathcal{I}(\boldsymbol{x};\boldsymbol{\theta})) = \frac{2}{\mid\Omega\mid}\int_{\Omega}\rho(\boldsymbol{x};\boldsymbol{\theta})\frac{\partial\rho(\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}d\boldsymbol{x} \tag{A.2}$$

The derivatives of the warped image are

$$\frac{\partial\mathcal{I}(\boldsymbol{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} = -\sum_{k=1}^{N}\pm_k\nabla\delta(\boldsymbol{x}-\boldsymbol{x}'_k(\boldsymbol{\theta}))\frac{\partial\boldsymbol{x}'_k(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}, \tag{A.3}$$

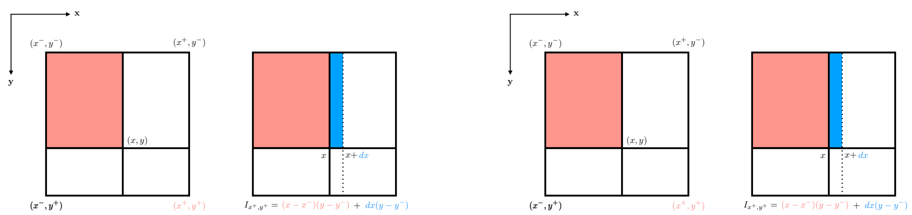
In the case of planar homography the events warping is computed as

$$\begin{aligned}
\boldsymbol{\theta} &= \left(\boldsymbol{\omega}^\top,\boldsymbol{v}^\top,\varphi,\psi\right)^\top\in\mathbb{R}^8 && \text{(A.4)}\\
\boldsymbol{x}'_k(\boldsymbol{\theta}) &= \begin{bmatrix}x'_{im} & y'_{im} & 1\end{bmatrix}^\top = \begin{bmatrix}x'/z' & y'/z' & 1\end{bmatrix}^\top && \text{(A.5)}\\
\bar{\boldsymbol{x}}'_k(\boldsymbol{\theta}) &= \begin{bmatrix}x' & y' & z'\end{bmatrix}^\top\\
&= \boldsymbol{P}\left(\boldsymbol{R}(t)^\top\left(\boldsymbol{I}+\boldsymbol{T}(t)\boldsymbol{n}^\top/d\right)\right)^{-1}\boldsymbol{x}_k\\
&= \boldsymbol{P}\left(\boldsymbol{I}+\boldsymbol{v}t\boldsymbol{n}^\top\right)^{-1}\boldsymbol{R}(t)\boldsymbol{x}_k, && \text{(A.6)}
\end{aligned}$$

where $\boldsymbol{P}$ is the projection matrix from the camera frame to the world frame or the map. Also, for simplicity of notation we assume $d=1$, and denote $\boldsymbol{P}\left(\boldsymbol{I}+\boldsymbol{v}t\boldsymbol{n}^\top\right)^{-1}$ as $\boldsymbol{P}_{\boldsymbol{v}}$. Since t usually spans a small temporal window, we can simplify the derivative with respect to angular velocity as

$$\begin{aligned}
\boldsymbol{R}(t) &= \exp(\boldsymbol{\omega}^\wedge t)\approx\boldsymbol{I}+\boldsymbol{\omega}^\wedge t && \text{(A.7)}\\
\frac{\partial\bar{\boldsymbol{x}}'_k}{\partial\boldsymbol{\omega}} &= -\boldsymbol{P}_{\boldsymbol{v}}t\boldsymbol{x}_k^\wedge. && \text{(A.8)}
\end{aligned}$$

The above equation makes use of the equivalence $\boldsymbol{\omega}\times\boldsymbol{x}_k = -\boldsymbol{x}_k\times\boldsymbol{\omega}$.



(a) The intensity at a pixel location is the area of the rectangle spanned by the opposite pixel and the event location

(b) Intensity change after an infinitesimal movement of the event

Figure A.1: Bilear voting.

The derivative with respect to the linear velocity is

$$\frac{\partial \bar{\boldsymbol{x}}'_k}{\partial \boldsymbol{v}} = -\frac{t\boldsymbol{n}^\top \boldsymbol{R}(t)\boldsymbol{x}_k}{\boldsymbol{n}^\top \boldsymbol{v}t + 1}\boldsymbol{P}_{\boldsymbol{v}} \tag{A.9}$$

$$eu \tag{A.10}$$

Similarly we have

$$\frac{\partial \bar{\boldsymbol{x}}'_k}{\partial \boldsymbol{n}} = -\frac{t\boldsymbol{v}^\top \boldsymbol{R}(t)\boldsymbol{x}_k}{\boldsymbol{n}^\top \boldsymbol{v}t + 1}\boldsymbol{P}_{\boldsymbol{v}} \tag{A.11}$$

and

$$\frac{\partial \bar{\boldsymbol{x}}'_k}{\partial (\varphi;\psi)} = \frac{\partial \bar{\boldsymbol{x}}'_k}{\partial \boldsymbol{n}}\frac{\partial \boldsymbol{n}}{\partial (\varphi;\psi)}, \tag{A.12}$$

with

$$\frac{\partial \boldsymbol{n}}{\partial (\varphi;\psi)} = \begin{bmatrix} -\sin\varphi\sin\psi & \cos\varphi\cos\psi \\ \cos\varphi\sin\psi & \sin\varphi\cos\psi \\ 0 & -\sin\psi \end{bmatrix} \tag{A.13}$$

With the above quantities

$$\frac{\partial \bar{\boldsymbol{x}}'_k}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial \bar{\boldsymbol{x}}'_k}{\partial \boldsymbol{\omega}} & \frac{\partial \bar{\boldsymbol{x}}'_k}{\partial \boldsymbol{v}} & \frac{\partial \bar{\boldsymbol{x}}'_k}{\partial \varphi} & \frac{\partial \bar{\boldsymbol{x}}'_k}{\partial \psi} \end{bmatrix} \in \mathbb{R}^{3\times 8}$$

we can compute the gradient $\frac{\partial \boldsymbol{x}'_k}{\partial \boldsymbol{\theta}}$ as

$$\frac{\partial \boldsymbol{x}'_k}{\partial \boldsymbol{\theta}} = \frac{\partial \bar{\boldsymbol{x}}'_k}{\partial \boldsymbol{\theta}}\frac{1}{z'} - \bar{\boldsymbol{x}}'_k\frac{\partial z'}{\partial \boldsymbol{\theta}}\frac{1}{z'^2} \tag{A.14}$$

Theoretically, the derivative of the Dirac delta $\nabla\delta(\boldsymbol{x})$ should be evaluated at each pixel location (240×180 for DAVIS) for each event. However, since we are using bilinear voting, a infinitesimal change $d\boldsymbol{x}$ at an event location $\boldsymbol{x}$ will only affect the derivative evaluated at the four neighboring pixels of $\boldsymbol{x}$. An illustration is shown in fig. A.1

The detailed algorithms for the evaluation of the Dirac delta as well as its derivative is shown below:

Input : *inactive locations* of last update $L_{k-1}(\boldsymbol{u})$
inactive reliability of last update$R_{k-1}(\boldsymbol{u})$
current camera pose $T_{d,g}$
Output: Updated *inactive locations* $L_k(\boldsymbol{u})$ and *inactive reliability* $R_k(\boldsymbol{u})$

Remap $L_{k-1}(\boldsymbol{u})$ to current pose and update $R_{k-1}(\boldsymbol{u})$ to get $L_k(\boldsymbol{u}), R_k(\boldsymbol{u})$
foreach *voxel block VB requested during swap-in* **do**
  foreach *voxel in VB* **do**
    get voxel global location $\boldsymbol{x}_{v,g} = h.\boldsymbol{pos} + \boldsymbol{x}_{v,VB}$
    forward project v to image plane with $\boldsymbol{u} = \pi\left(KT_{d,g}^{-1}\tilde{\boldsymbol{x}}_{v,g}\right)$
    if $\boldsymbol{u}$ *in current image* **then**
      if $L_k(\boldsymbol{u})$ *empty* **then**
        $L_k(\boldsymbol{u}) = \boldsymbol{x}_{v,g} + \mu.sdf.\boldsymbol{r}$
        $R_k(\boldsymbol{u}) = v.reliability$
      else if $v.reliability > R_k(\boldsymbol{u})$ **then**
        $v.reliability = L_k(\boldsymbol{u})$
      end
    end
  end
end

Algorithm 1: sIncremental Raycasting

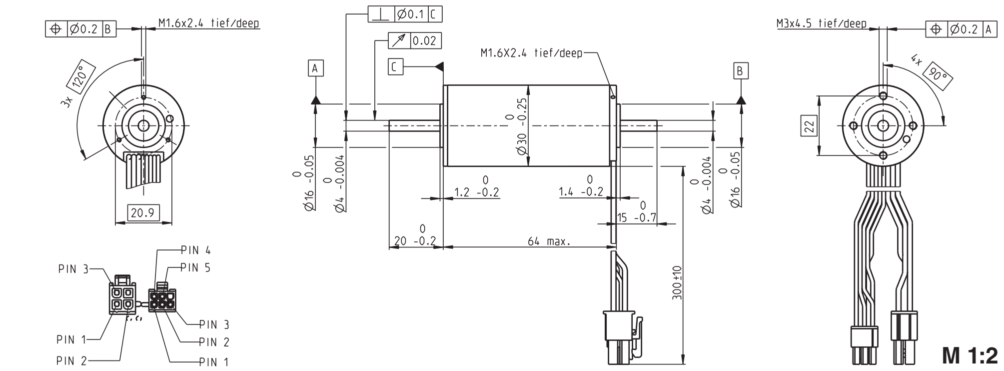
However, a general nonlinear optimization is usually very hard, especially at bundle adjustment stage he jacobian matrix only measures the local grad

Appendix B

Datasheets

EC-max 30 ∅30 mm, brushless, 60 Watt

maxon *EC*-max



M 1:2

Stock program
Standard program
Special program (on request)

Part Numbers

Motor Data		272762	272763	272764	272765
Values at nominal voltage					
1 Nominal voltage	V	12	24	36	48
2 No load speed	rpm	7980	9340	9490	9350
3 No load current	mA	302	191	130	95.4
4 Nominal speed	rpm	6590	8040	8270	8130
5 Nominal torque (max. continuous torque)	mNm	63.6	60.7	63.7	64.1
6 Nominal current (max. continuous current)	A	4.72	2.66	1.88	1.4
7 Stall torque	mNm	381	458	522	519
8 Starting current	A	26.8	18.8	14.5	10.7
9 Max. efficiency	%	80	81	82	82
Characteristics					
10 Terminal resistance phase to phase	Ω	0.447	1.27	2.48	4.49
11 Terminal inductance phase to phase	mH	0.049	0.143	0.312	0.573
12 Torque constant	mNm/A	14.2	24.3	35.9	48.6
13 Speed constant	rpm/V	672	393	266	197
14 Speed/torque gradient	rpm/mNm	21.2	20.6	18.4	18.2
15 Mechanical time constant	ms	4.86	4.73	4.21	4.17
16 Rotor inertia	gcm^2	21.9	21.9	21.9	21.9

Specifications

Thermal data	
17 Thermal resistance housing-ambient	7.4 K/W
18 Thermal resistance winding-housing	0.5 K/W
19 Thermal time constant winding	2.76 s
20 Thermal time constant motor	1000 s
21 Ambient temperature	-40…+100°C
22 Max. permissible winding temperature	+155°C
Mechanical data (preloaded ball bearings)	
23 Max. permissible speed	15000 rpm
24 Axial play at axial load < 6.0 N	0 mm
> 6.0 N	0.14 mm
25 Radial play	preloaded
26 Max. axial load (dynamic)	5 N
27 Max. force for press fits (static)	98 N
(static, shaft supported)	1300 N
28 Max. radial loading, 5 mm from flange	25 N
Other specifications	
29 Number of pole pairs	1
30 Number of phases	3
31 Weight of motor	305 g

Values listed in the table are nominal.

Connection motor (Cable AWG 20)

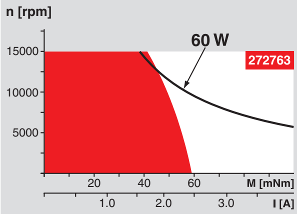
red	Motor winding 1	Pin 1
black	Motor winding 2	Pin 2
white	Motor winding 3	Pin 3
	N.C.	Pin 4
Connector	**Part number**	
Molex	39-01-2040	

Connection Sensors (Cable AWG 26)

yellow	Hall sensor 1	Pin 1
brown	Hall sensor 2	Pin 2
grey	Hall sensor 3	Pin 3
blue	GND	Pin 4
green	V_{Hall} 3…24 VDC	Pin 5
	N.C.	Pin 6
Connector	**Part number**	
Molex	430-25-0600	

Wiring diagram for Hall sensors see p. 35

Operating Range



Comments

Continuous operation
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring).

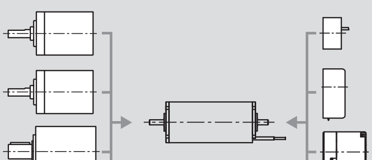
Assigned power rating

maxon Modular System Overview on page 20 - 25

Planetary Gearhead
∅32 mm
8.0 Nm
Page 266

Koaxdrive
∅32 mm
1.0 - 4.5 Nm
Page 268

Planetary Gearhead
∅42 mm
3 - 15 Nm
Page 271



Encoder MR
500/1000 CPT,
3 channels
Page 302

Encoder HEDL 5540
500 CPT,
3 channels
Page 308

Brake AB 20
24 VDC
0.1 Nm
Page 346

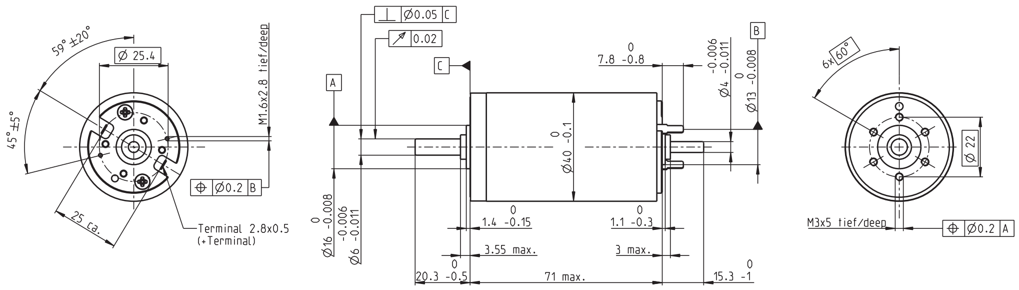
Recommended Electronics:

ESCON 36/3 EC	Page 320
ESCON 50/5, Module 50/5	321
ESCON 70/10	321
DECS 50/5	324
DEC Module 24/2	325
DEC Module 50/5	325
EPOS2 24/5, 50/5	331
EPOS2 P 24/5	334
EPOS3 70/10 EtherCAT	337
Notes	**24**

RE 40 ∅40 mm, Precious Metal Brushes, 25 Watt

NEW

maxon DC motor



M 1:2

- Stock program
- Standard program
- Special program (on request)

Part Numbers

Motor Data			448588	448589	448590	448591	448592
	Values at nominal voltage						
1	Nominal voltage	V	9	18	24	42	48
2	No load speed	rpm	2850	2850	2780	2920	2690
3	No load current	mA	49.7	24.8	18.1	11	8.62
4	Nominal speed	rpm	2610	2600	2480	2640	2410
5	Nominal torque (max. continuous torque)	mNm	87.8	87.8	88.2	87.6	87.6
6	Nominal current (max. continuous current)	A	2.96	1.48	1.09	0.65	0.524
7	Stall torque	mNm	873	956	794	895	818
8	Starting current	A	29	15.9	9.66	6.53	4.81
9	Max. efficiency	%	92	92	92	92	92
	Characteristics						
10	Terminal resistance	Ω	0.311	1.14	2.49	6.43	9.97
11	Terminal inductance	mH	0.0824	0.33	0.613	1.7	2.62
12	Torque constant	mNm/A	30.2	60.3	82.2	137	170
13	Speed constant	rpm/V	317	158	116	69.7	56.2
14	Speed / torque gradient	rpm/mNm	3.27	2.98	3.51	3.27	3.3
15	Mechanical time constant	ms	4.85	4.29	4.36	4.14	4.13
16	Rotor inertia	gcm^2	142	137	119	121	120

Specifications

Thermal data

17	Thermal resistance housing-ambient	4.65 K/W
18	Thermal resistance winding-housing	1.93 K/W
19	Thermal time constant winding	41.5 s
20	Thermal time constant motor	809 s
21	Ambient temperature	-20…+85°C
22	Max. permissible winding temperature	+100°C

Mechanical data (ball bearings)

23	Max. permissible speed	3330 rpm
24	Axial play	0.05 - 0.15 mm
25	Radial play	0.025 mm
26	Max. axial load (dynamic)	5.6 N
27	Max. force for press fits (static)	110 N
	(static, shaft supported)	1200 N
28	Max. radial loading, 5 mm from flange	28 N

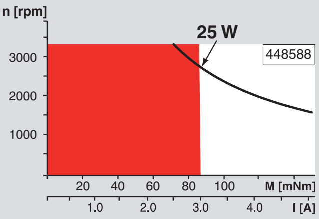
Other specifications

29	Number of pole pairs	1
30	Number of commutator segments	13
31	Weight of motor	480 g

Values listed in the table are nominal.
Explanation of the figures on page 71.

Option
Preloaded ball bearings

Operating Range



Comments

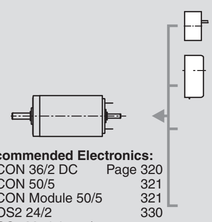
Continuous operation
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring).

Assigned power rating

maxon Modular System

Overview on page 20 - 25



Encoder MR
256 - 1024 CPT,
3 channels
Page 303

Encoder HED_ 5540
500 CPT,
3 channels
Page 305/307

Recommended Electronics:

ESCON 36/2 DC	Page 320
ESCON 50/5	321
ESCON Module 50/5	321
EPOS2 24/2	330
EPOS2 Module 36/2	330
EPOS2 24/5	331
EPOS2 50/5	331
EPOS2 P 24/5	334
EPOS3 70/10 EtherCAT	337
Notes	**22**