Conference Paper

Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints

Author(s):
Scaramuzza, Davide; Fraundorfer, Friedrich; Pollefeys, Marc; Siegwart, Roland

Publication Date:
2009

Permanent Link:
https://doi.org/10.3929/ethz-a-010035851

Originally published in:
http://doi.org/10.1109/ICCV.2009.5459294

Rights / License:
In Copyright - Non-Commercial Use Permitted
Absolute Scale in Structure from Motion from a Single Vehicle Mounted Camera by Exploiting Nonholonomic Constraints

Davide Scaramuzza\textsuperscript{1}, Friedrich Fraundorfer\textsuperscript{2}, Marc Pollefeys\textsuperscript{2}, Roland Siegwart\textsuperscript{1}

\textsuperscript{1} Autonomous Systems Lab
\textsuperscript{2} Computer Vision and Geometry Group
ETH Zurich

Abstract

In structure-from-motion with a single camera it is well known that the scene can be only recovered up to a scale. In order to compute the absolute scale, one needs to know the baseline of the camera motion or the dimension of at least one element in the scene. In this paper, we show that there exists a class of structure-from-motion problems where it is possible to compute the absolute scale completely automatically without using this knowledge, that is, when the camera is mounted on wheeled vehicles (e.g. cars, bikes, or mobile robots). The construction of these vehicles puts interesting constraints on the camera motion, which are known as “nonholonomic constraints” The interesting case is when the camera has an offset to the vehicle’s center of motion. We show that by just knowing this offset, the absolute scale can be computed with a good accuracy when the vehicle moves. We give a mathematical derivation and provide experimental results on both simulated and real data. To our knowledge this is the first time nonholonomic constraints of wheeled vehicles are used to estimate the absolute scale. We believe that the proposed method can be useful in those research areas involving visual odometry and mapping with vehicle mounted cameras.

1. Introduction

Visual odometry (also called structure from motion) is the problem of recovering the motion of a camera from the visual input alone. This can be done by using single cameras (perspective or omnidirectional) \cite{2, 10}, stereo cameras \cite{6}, or multi-camera systems \cite{1}. The advantage of using more than one camera is that both the motion and the 3D structure can be computed directly in the absolute scale when the distance between the cameras is known. Furthermore, the cameras not necessarily need to have an overlapping field of view, as shown in \cite{1}. Conversely, when using a single camera the absolute scale must be computed in other ways, like by measuring the motion baseline or the size of an element in the scene \cite{2}, or by using other sensors like IMU and GPS \cite{7}.

In the case of a single camera mounted on a vehicle, the camera follows the movement of the vehicle. Most wheeled vehicles (e.g. car, bike, mobile robot) possess an instantaneous center or rotation, that is, there exists a point around which each wheel of the vehicle follows a circular course \cite{11}. For instance, for car-like vehicles the existence of this point is insured by the Ackerman steering principle (Fig. 2). This property assures that the vehicle undergoes rolling motion, i.e. without slippage. Accordingly, the motion of the vehicle can be locally described by circular motion. As we will show in the paper, this property puts interesting constraints on the camera motion. Depending on the position of the camera on such a vehicle, the camera can undergo exactly the same motion or deviate from it. The interesting case is when the camera has an offset to the vehicle center of motion (see Fig. 1). By just knowing this offset and the camera relative motion from the point correspondences, the absolute scale can be computed. The key concept of
our new method is that, because of the Ackerman steering model, the different motions of camera and vehicle can be computed from the same camera measurements. Then, the difference between them can be used to compute the absolute scale. In this paper, we describe the method to compute absolute scale for a vehicle moving in a plane. We give a minimal solution as well as a least-squares solution to the absolute scale. In addition, we also present and efficient algorithm that can cope with outliers.

The recent street level mapping efforts of various companies and research centers make the proposed approach very interesting. In these cases the cars are usually equipped with a single omni-directional camera and with our novel method it would be possible to compute the absolute scale of the recovered map.

This paper is organized as follows. Section 2 reviews the related work. Section 3 explains the motion model of wheeled vehicles. Section 4 provides the equations for computing the absolute scale. Section 5 explains how to detect the circular motion. Finally, sections 6 and 7 present the experimental results and conclusions.

2. Related work

The standard way to get the absolute scale in motion estimation is the use of a stereo setup with known baseline. A very well working approach in this fashion has been demonstrated by Nister et al. [6]. The fields of views of the two cameras were overlapping and motion estimation was done by triangulating feature points, tracking them, and estimating new poses from them. Other approaches using stereo setups are described in [3, 4] and can be traced back to as early as [5]. A recent approach from Clipp et al. [1] relaxed the need of overlapping stereo cameras. They proposed a method for motion estimation including absolute scale from two non-overlapping cameras. From independently tracked features in both cameras and with known baseline, full 6DOF motion could be estimated. In their approach the motion up to scale was computed from feature tracks in one camera. The remaining absolute scale could then be computed from one additional feature track in the other camera.

For the case of single cameras, some prior knowledge about the scene has been used to recover the absolute scale. Davison et al. [2] used a pattern of known size for both initializing the feature locations and computing the absolute scale in 6DOF visual odometry. Scaramuzza et al. [10] used the distance of the camera to the plane of motion and feature tracks from the ground plane to compute the absolute scale in a visual odometry system for ground vehicle applications.

In this paper, we propose a completely novel approach to compute the absolute scale from a single camera mounted on a vehicle. Our method exploits the constraint imposed by nonholonomic wheeled vehicles, that is, their motion can be locally described by circular motion.

3. Motion model of nonholonomic vehicles

A vehicle is said to be nonholonomic if its controllable degrees of freedom are less than its total degrees of freedom [11]. An automobile is an example of a nonholonomic vehicle. The vehicle has three degrees of freedom, namely its position and orientation in the plane. Yet it has only two controllable degrees of freedom, which are the acceleration and the angle of the steering. A car’s heading (the direction in which it is traveling) must remain aligned with the orientation of the car, or $180^\circ$ from it if the car is going backward. It has no other allowable direction. The nonholonomicity of a car makes parking and turning in the road difficult. Other examples of nonholonomic wheeled vehicles are bikes and most mobile robots.

The nonholonomicity reveals an interesting property of the vehicle’s motion, that is, the existence of an Instantaneous Center of Rotation (ICR). Indeed, for the vehicle to exhibit rolling motion without slipping, a point must exist around which each wheel of the vehicle follows a circular course. The ICR can be computed by intersecting all the roll axes of the wheels (see Fig. 2). For cars the existence of the ICR is ensured by the Ackermann steering principle [11]. This principle ensures a smooth movement of the vehicle by applying different steering angles to the left and right front wheel while turning. This is needed as all the four wheels move in a circle on four different radii around the ICR (Fig. 2). As the reader can perceive, every point of the vehicle and any camera installed on it undergoes locally planar circular motion. Straight motion can be represented along a circle of infinite radius of curvature.

Let us now derive the mathematical constraint on the vehicle motion. Planar motion is described by three parameters, namely the rotation angle $\theta$, the direction of translation $\varphi_v$, and the length $\rho$ of the translation vector (Fig. 3(a)). However, for the particular case of circular motion and when the vehicle’s origin is chosen along the nonsteering axle as in Fig. 3(a), we have the interesting property that $\varphi_v = \theta / 2$. This property can be trivially verified by trigonometry. Accordingly, if the camera reference frame coincides with the car reference frame, we have that the camera must verify the same constraint $\varphi_c = \theta / 2$. However, this constraint is no longer valid if the camera has an offset $L$ with the vehicle’s origin as shown in Fig. 3(b). In this case, as we will show in the next section, a more complex constraint exists which relates $\varphi_c$ to $\theta$ through the offset $L$ and the vehicle’s displacement $\rho$. Since $L$ is constant and can be measured very accurately, we will show that it is then possible to estimate $\rho$ (in the absolute scale) by just knowing $\varphi_c$ and $\theta$ from point correspondences.

\[ \text{DOF} = \text{Degrees Of Freedom} \]
the vehicle’s origin, we have still no additional rotation between them. The camera is denoted by $\mathbf{P}$. Both coordinate systems are aligned so that there is a constraint. When camera and vehicle reference systems coincide (see also Fig. 3).

Figure 3 shows the camera and vehicle coordinate systems. Both coordinate systems are aligned so that there is no additional rotation between them. The camera is denoted by $\mathbf{P}_1$ and it is located at $C_1 = [0, 0, L]$ in the vehicle coordinate system. The camera matrix $\mathbf{P}_1$ is therefore

$$
\mathbf{P}_1 = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & -L \\
\end{bmatrix}
$$

(1)

The camera $\mathbf{P}_1$ and the vehicle now undergo the following circular motion denoted by the rotation $\mathbf{R}$ and the translation $\mathbf{T}$ (see also Fig. 3).

$$
\mathbf{R} = \begin{bmatrix}
\cos(\theta) & 0 & -\sin(\theta) \\
0 & 1 & 0 \\
\sin(\theta) & 0 & \cos(\theta) \\
\end{bmatrix}
$$

(2)

$$
\mathbf{T} = \rho \begin{bmatrix}
\sin\left(\frac{\phi}{2}\right) \\
0 \\
\cos\left(\frac{\phi}{2}\right) \\
\end{bmatrix}
$$

(3)

The transformed camera $\mathbf{P}_2$ is then

$$
\mathbf{P}_2 = [\mathbf{R}_2 \, \mathbf{t}_2] = \mathbf{P}_1 \begin{bmatrix}
R & -RT \\
0 & 1 \\
\end{bmatrix}
$$

(4)

To compute the motion between the two cameras $\mathbf{P}_2$ and $\mathbf{P}_1$ the camera $\mathbf{P}_2$ can be expressed in the coordinate system of $\mathbf{P}_1$. Let us denote it by $\mathbf{P}'_2$. $\mathbf{P}'_2 = [\mathbf{R}'_2 \, \mathbf{t}'_2] = \mathbf{P}_2 \begin{bmatrix}
P_1 & 0 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}^{-1}

(5)

The rotation part $\mathbf{R}'_2$ equals $\mathbf{R}_2$ (which equals $\mathbf{R}$) and the translation part $\mathbf{t}'_2$ is

$$
\mathbf{t}'_2 = \begin{bmatrix}
\rho \sin\left(\frac{\phi}{2}\right) - L \sin(\theta) \\
L \cos(\theta) - \rho \cos\left(\frac{\phi}{2}\right) - L \\
\end{bmatrix}
$$

(6)

Then, the essential matrix $\mathbf{E}$ for our setup describing the relative motion from camera $\mathbf{P}_1$ to $\mathbf{P}_2$ is defined as $\mathbf{E} = [\mathbf{t}'_2 \times \mathbf{R}'_2]$ and can be written as:

$$
\mathbf{E} = \begin{bmatrix}
L - \rho \cos\left(\frac{\phi}{2}\right) - L \cos(\theta) & L + \rho \cos\left(\frac{\phi}{2}\right) - L \cos(\theta) & \rho \sin\left(\frac{\phi}{2}\right) + L \sin(\theta) \\
0 & \rho \sin\left(\frac{\phi}{2}\right) - L \sin(\theta) & \rho \sin\left(\frac{\phi}{2}\right) - L \sin(\theta) \\
0 & 0 & 1 \\
\end{bmatrix}
$$

(7)

Finally, the essential matrix can also be expressed in terms of the absolute distance $\lambda$ between the two camera centers, and the camera relative motion $(\theta, \phi_c)$. Thus, using the previous expression of $\mathbf{R}'_2$ and $\mathbf{t}'_2$ (but now in terms of $\lambda$, $\theta$, and $\phi_c$) we obtain:

$$
\mathbf{E} = \lambda \begin{bmatrix}
0 & \cos(\theta - \phi_c) & 0 \\
-\cos(\phi_c) & 0 & \sin(\phi_c) \\
0 & \sin(\theta - \phi_c) & 0 \\
\end{bmatrix}
$$

(8)

These two expressions for $\mathbf{E}$ will be used in the next sections.

4.2. Computing $\rho$ and $\lambda$ from rotation and translation angles

To recap, the parameter $\rho$ is the absolute distance between the two vehicle positions (Fig. 3(a)), while $\lambda$ is the absolute distance between the two camera centers which is $\lambda = ||\mathbf{t}'_2||$ (Fig. 3(b)).

It is convenient to be able to express $\rho$ and $\lambda$ in terms of the rotation angle $\theta$ and the directional angle $\phi_c$ of the camera translation vector because these parameters can be estimated from feature correspondences. For this we equate the camera center $C_2' = -\mathbf{R}'_2 \mathbf{t}'_2$ with a parameterization in
\( \varphi_c \). From this we can get the equations for \( \rho \) and \( \lambda \) in terms of \( \theta \) and \( \varphi_c \).

\[
\rho = \frac{L \sin(\varphi_c) + L \sin(\theta - \varphi_c)}{-\sin\left(\frac{\theta}{2} - \varphi_c\right)} \quad (9)
\]

\[
\lambda = \frac{-2L \sin\left(\frac{\theta}{2}\right)}{\sin\left(\frac{\theta}{2} - \varphi_c\right)} \quad (10)
\]

Note, expressions (9) and (10) are exactly the core of this paper, that is, we can actually compute the absolute distance between the vehicle/camera centers as a function of the camera offset \( L \) and the camera relative motion \((\theta, \varphi_c)\). In the next subsection we will give a minimal and a least-square solution to compute the \((\theta, \varphi_c)\) directly from a set of point correspondences. Finally, note that (9) and (10) are valid only if \( \theta \neq 0 \). Thus, we can only estimate the absolute scale if the vehicle rotates. The accuracy on the absolute scale estimates will be evaluated in Section 6.1.

Observe that we can also write an equation for \( L \). This allows us to compute the offset of the camera from the rear axis of the vehicle from ground truth data (GPS, wheel odometry, etc.), i.e. to calibrate the camera to the vehicle coordinate frame. By solving (9) with respect to \( L \) we have:

\[
L = \rho \frac{-\sin\left(\frac{\theta}{2} - \varphi_c\right)}{\sin(\varphi_c) + \sin(\theta - \varphi_c)} \quad (11)
\]

### 4.3. Least-squares solution: the 3-point algorithm

In this section, we provide a least-squares solution to compute \( \theta \) and \( \varphi_c \) from a set of good feature correspondences. Two corresponding points \( p = (x, y, z)^T \) and \( p' = (x', y', z')^T \) must fulfill the epipolar constraint

\[
p^T E p = 0 \quad (12)
\]

Using the expression (8) of the essential matrix, the epipolar constraint expands to:

\[
-xy' \cos(\varphi_c) + yx' \cos(\theta - \varphi_c) + 
y\sin(\varphi_c) + y'z' \sin(\theta - \varphi_c) = 0. \quad (13)
\]

Given \( m \) image points, \( \theta \) and \( \varphi_c \) can be computed indirectly using singular value decomposition of the coefficient matrix \([xy', yx', y\sin(\varphi_c), y'z', yz']\) being \([h_1, h_2, h_3, h_4] \) the unknown vector which is defined by:

\[
\begin{align*}
h_1 &= -\cos(\varphi_c), \\
h_2 &= \cos(\theta - \varphi_c), \\
h_3 &= \sin(\varphi_c), \\
h_4 &= \sin(\theta - \varphi_c).
\end{align*} \quad (14)
\]

4.4. Minimal solution: non-linear 2-point algorithm

As shown by Eq. (13), the epipolar constraint can be reduced to a non-linear equation \( f(\theta, \varphi_c) = 0 \) which can be solved by Newton’s iterative method. This method is based on a first order Taylor expansion of \( f \), that is,

\[
f(\theta, \varphi_c) \approx f(\theta_0, \varphi_{c0}) + J_f(\theta_0, \varphi_{c0}) \begin{bmatrix} \theta - \theta_0 \\ \varphi - \varphi_{c0} \end{bmatrix} \quad (15)
\]

where \( f(\theta_0, \varphi_{c0}) \) can be computed from (13) and the Jacobian \( J_f(\theta_0, \varphi_{c0}) \) can be written as:

\[
J_f(\theta_0, \varphi_{c0}) = 
\begin{bmatrix}
-xy' \sin(\theta_0 - \varphi_{c0}) & + y'z' \cos(\theta_0 - \varphi_{c0}) \\
xy' \sin(\varphi_{c0}) & + y'z' \sin(\theta_0 - \varphi_{c0}) - yz' \cos(\theta_0 - \varphi_{c0}) + yz' \cos(\varphi_{c0})
\end{bmatrix} \quad (16)
\]

Newton’s method is an iterative method which starts from an initial seed and converges to the solution through successive approximations which are computed as:

\[
\begin{bmatrix}
\theta_{i+1} \\
\varphi_{ci+1}
\end{bmatrix} = J_f(\theta_i, \varphi_{ci})^{-1} f(\theta_i, \varphi_{ci}) + 
\begin{bmatrix}
\theta_i \\
\varphi_{ci}
\end{bmatrix} \quad (17)
\]

In all the experimental results we had convergence by taking the point \((\theta_0, \varphi_{c0}) = (0, 0)\) as initial seed. The algorithm converged very quickly (3-4 iterations). Since only two unknowns are determined, two is the minimum number of matches required by this algorithm to compute the solution.

A comparison of the performance between the linear 3-point and the non-linear 2-point algorithm is given in the experimental section 6.1.

5. Circular motion detection

The equations for absolute scale estimation give only correct results if the motion is circular. Thus we have to identify sections of circular motion in a camera path prior to computing the absolute scale. For circular motion \( \theta/2 - \varphi_v = 0 \), the idea is therefore to look for motion that satisfies this condition. The first step is to compute \( \varphi_v \) from the camera motion \((\theta, \varphi_c)\). \( \varphi_v \) is a function of \( L, \rho, \varphi_c \), but \( \rho \) is unknown. We therefore propose to search for \( \varphi_v(L, \rho, \varphi_c) \) that minimizes the criterion for circular motion \( \theta/2 - \varphi_v \) by varying \( \rho \). This is a 1D optimization over \( \rho \) that can be solved with Newton’s method. The optimization converges very quickly and returns \( \varphi_v \) that minimizes our circular motion condition. If the motion is circular \( |\theta/2 - \varphi_v| \) gets very small; for non-circular motion the condition is not exactly satisfied. To distinguish between circular and non-circular motion we introduce the threshold \( \text{thresh}_{circ} \). A motion is classified as circular if \( |\theta/2 - \varphi_v| < \text{thresh}_{circ} \) and non-circular otherwise.
Figure 4. The relative error % of the absolute scale estimate as a function of the rotation angle $\theta$. Comparison between the linear 3-point method (circles) and the non-linear 2-point method (squares).

6. Experiments

6.1. Synthetic data

We investigated the performance of the algorithms in geometrically realistic conditions. In particular, we simulated a vehicle moving in urban canyons where the distance between the camera and facades is about 10 meters. We set the first camera at the origin and randomized scene points uniformly inside several different planes, which stand for the facades of urban buildings. We used overall 1600 scene points. The second camera was positioned according to the motion direction of the vehicle which moves along circular trajectories about the instantaneous center of rotation. Therefore, the position of the second camera was simulated according to the previous equations by taking into account the rotation angle $\theta$, the vehicle displacement $\rho$, and the offset $L$ of the camera from the vehicle’s origin. To make our analysis more general, we considered an omnidirectional camera (with the same model used in the real experiments), therefore the scene points are projected from all directions. Finally, we also simulated feature location errors by introducing a $\sigma = 0.3$ pixel Gaussian noise in the data. The image resolution was set to a $640 \times 480$ pixels.

In this experiment, we want to evaluate the accuracy of the estimated absolute scale as a function of the rotation angle $\theta$. As observed in equation (9), the estimate of the absolute scale $\rho$ from the camera relative motion is possible only for $\theta \neq 0$. Therefore, we can intuitively expect that the absolute scale accuracy increases with $\theta$. In this experiment, we performed many trials (one hundred) for different values of $\theta$ (varying from 0 up to 30 deg). The results shown in Fig. 4 are the average. As observed, the accuracy improves with $\theta$, with an error smaller than 5% for $\theta$ larger than 10 deg. The performance of the linear and non-linear algorithm are similar when $\theta > 10$ deg, while the non-linear method performs better for smaller $\theta$.

6.2. Real data

In this section we demonstrate the absolute scale computation on an image sequence acquired by a car equipped with an omnidirectional camera driving through a city in a 3Km tour. A picture of our vehicle (a Smart) is shown in Fig. 1. The omnidirectional camera is composed of a hyperbolic mirror (KAIDAN 360 One VR) and a digital color camera (SONY XCD-SX910, image size $1280 \times 960$ pixels). The camera was installed as shown in Fig. 1(b). The offset of the camera from the rear axle is $L=0.9$m. The camera system was calibrated using the toolbox from Scaramuzza [9, 8]. Images were taken at an average framerate of 10Hz at a vehicle speed ranging from 0 to 45km/h. In an initial step, up to scale motion estimation has been performed. We did this for the all 4000 frames of the dataset. In addition to the visual measurements, we have the wheel odometry measurements of the car. We will use the odometry measurements as baseline to which we compare our absolute scale values. Here it should be noted that the wheel odometry does not represent exactly the same measurements as our estimated absolute scale. The wheel odometry represents the length of the arc the wheels were following while the absolute scale represents the direct distance between the locations at which frames were captured. To identify sections of circular motion we look at the motion of neighboring frames. If the motion between neighboring frames is too small we look ahead to frames that are further out. In the experiments we maximally look ahead 15 frames. For each frame pair we check if it represents circular motion by checking if $|\theta/2 - \phi_v| < \text{thresh}_{\text{circ}}$ as described in section 5. The basic outline of the algorithm is described in Fig. 5. Fig. 6(a) shows a single curve from the path. The section apparently is partly a circular motion. It is quite reasonable if you look at it. In the picture, sections of cir-

![Figure 4. The relative error % of the absolute scale estimate as a function of the rotation angle $\theta$. Comparison between the linear 3-point method (circles) and the non-linear 2-point method (squares).](image)

![Figure 5. Outline of the absolute scale algorithm](image)
Circular motion are indicated by green dots. The section ends at the green circles. The sections were detected based on $|\theta/2 - \varphi_v| < \text{thresh}_\text{circ}$. For each detected section we computed the absolute scale and compared it to the wheel odometry. The difference was less than 30%. In the following we classify measurements with a difference less than 30% as correct and wrong otherwise. Fig. 6(b) shows the graph of the values $\theta$, $\theta/2$ and $\varphi_v$ for this section. It is clearly visible that it is largely a circular motion as $\theta/2$ and $\varphi_v$ are almost equal for large parts. At the end of the section the motion stops to be circular, which is also evident in Fig. 6(a) as the last part of the curve straightens. Fig. 7 shows $\theta/2$ and $\varphi_v$ for the full sequence. At around 50% of the sequence $\theta/2$ and $\varphi_v$ overlap, thus indicating circular motion. Multiple sections show a high turning angle $\theta$ (peaks) which can be used to compute the absolute scale accurately according to our previous synthetic results.

Results on the accuracy of the method and on the influence of the thresholds are shown in Table 1. Here we list the number of correctly computed scales, and the number of all detected circular motion sections. By tuning the thresholds $\theta_{\text{thresh}}$ and $\text{thresh}_\text{circ}$ the results can be optimized. $\text{thresh}_\text{circ}$ controls if a section of the path is classified as circular. The absolute scale of such a section will only be computed if the turning angle $\theta$ is larger than $\theta_{\text{thresh}}$. With a larger threshold on $\theta$ it is possible to reduce the number of wrong computed scales. With a threshold setting of 30° it was possible to remove all wrong estimates. With this setting 8 circular motion sections got detected and the absolute scale difference to the wheel odometry was below the threshold of 30%. The mean difference in this case was 20.6% (std. dev. 7.6%). This is a mean absolute difference of 2.3m (std. dev. 0.9m) which is a satisfying result. Note that the number of detected circular sections can be larger than the number of frames as we look at frame pairs.

Fig. 8 shows a plot of the full path. Sections of circular motion are shown as green dots. Circular motion appears not only in sharp turns but also at slight curves. Red circles show the sections where we computed the most accurate absolute scale measurements.

The results demonstrate that our method is able to properly detect sections of circular motion in a camera path and that it is possible to compute the absolute scale accurately. In our case the offset of $L = 0.9m$ is actually rather small and we would expect even better results with a larger offset.
<table>
<thead>
<tr>
<th>$\theta_{\text{thresh}}[^\circ]$</th>
<th># correct</th>
<th># detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>449</td>
<td>3057</td>
</tr>
<tr>
<td>10</td>
<td>65</td>
<td>153</td>
</tr>
<tr>
<td>20</td>
<td>21</td>
<td>36</td>
</tr>
<tr>
<td>30</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

(a) $\text{thresh}_{\text{circ}} = 10^{-4}$

<table>
<thead>
<tr>
<th>$\theta_{\text{thresh}}[^\circ]$</th>
<th># correct</th>
<th># detected</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>649</td>
<td>10803</td>
</tr>
<tr>
<td>10</td>
<td>170</td>
<td>2267</td>
</tr>
<tr>
<td>20</td>
<td>74</td>
<td>789</td>
</tr>
<tr>
<td>30</td>
<td>28</td>
<td>168</td>
</tr>
</tbody>
</table>

(b) $\text{thresh}_{\text{circ}} = 0.1$

Table 1. Table shows the number of detected circular motions and the number of correct estimated absolute scales (within 30% of wheel odometry measurements) for different thresholds of $\theta_{\text{thresh}}$. A threshold on $\theta$ is very effective in removing inaccurate estimates, i.e., only motions with large $\theta$ give accurate estimates.

7. Conclusion

In this paper, we have shown that the nonholonomic constraints of wheeled vehicles (e.g., car, bike, differential drive robot) make it possible to estimate the absolute scale in structure from motion from a single vehicle mounted camera. We have shown that this can be achieved whenever the camera has an offset to the vehicle center of motion and when the vehicle is turning. This result is made possible by the fact that the vehicle undergoes locally planar circular motion.

Our experiments show that a good accuracy can be achieved even with a small offset like 0.9 m, although a larger offset would increase the accuracy. The camera does not need to be placed at a specific place. This allows us to process data that has already been captured, as most cameras will be placed off-axis. We also showed that our method successfully detects sections of circular motion in a camera path. The experiments showed that actually a large amount of vehicle motion is in fact circular. Future work would include improvement of the circular motion detection as this is very important to create a robust algorithm. This could then be used to reduce the unavoidable scale drift of structure-from-motion system. If an absolute scale can be computed reliably every hundred frames or so this will stabilize scale over time.

References


[9] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easy calibrating omnidirectional cameras. In IEEE In-