# Absolute Scale in Structure from Motion from a Single Vehicle Mounted Camera by Exploiting Nonholonomic Constraints

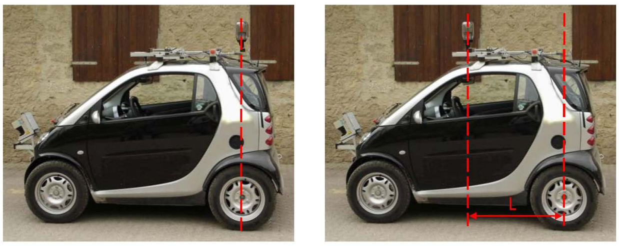
Davide Scaramuzza[1,*], Friedrich Fraundorfer[2], Marc Pollefeys[2], Roland Siegwart[1]
[1] Autonomous Systems Lab
[2] Computer Vision and Geometry Group
ETH Zurich

## 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 turns. We give a mathematical derivation and provide experimental results on both simulated and real data over a large image dataset collected during a 3 Km path. 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) [2, 14], stereo cameras [8], or multi-camera systems [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 [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 [2], or by using other sensors like IMU and GPS [10].



(a) (b)

Figure 1. If the camera is located on the vehicle's non steering axle, the rotation and translation of both the camera and the car are exactly the same (a). If the camera is mounted with an offset to the axle, rotation and translation of camera and car are different (b). While case (a) can be used to simplify motion estimation (see our previous work [12]), case (b) can be used to compute the absolute scale from a single camera.

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 [15]. 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

[*] This work was supported by the European grants FP6-IST-1-045350 (Robots@Home) and FP6-IST-027140 (BACS)

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 when the vehicle turns.

The recent efforts of various companies and research centers on street level mapping make the proposed approach very interesting. In these cases the cars are usually equipped with a single omni-directional camera (see as an example Google cars) 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 describes two algorithms for estimating the motion under planar assumption. 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. [8]. 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 [5, 6] and can be traced back to as early as [7]. 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 [1] 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. [14] 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. The geometry of circular motion has been deeply studied in [3, 4] where the application was 3D shape recovery using a turntable. However, the theory explained in this document and the application to the absolute scale computation are completely novel.

Finally, observe that this paper is the follow-up of our previous paper [12], where we used the circular motion constraint to design the two most efficient algorithms for removing the outliers of the feature matching process: 1-point RANSAC and Histogram Voting. However the present document has been conceived to be stand alone. Indeed, feature correspondences are assumed to be provided already.

## 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 [15]. 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 [15]. 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 non-steering 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

[1] DOF = Degrees Of Freedom

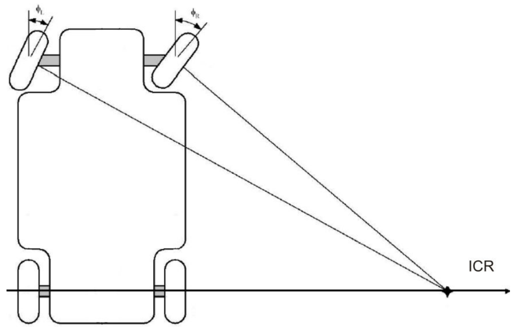
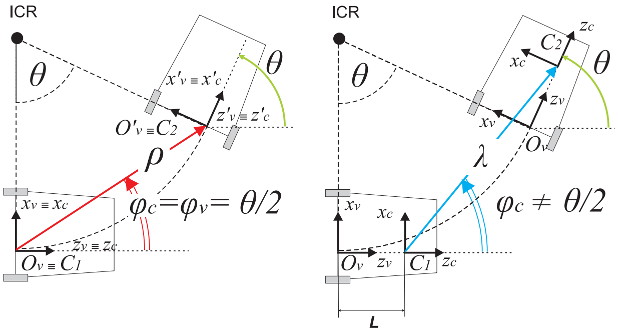


Figure 2. General Ackermann steering principle



(a) $L = 0$ (b) $L \neq 0$

Figure 3. Camera and vehicle motion under circular motion constraint. When camera and vehicle reference systems coincide $\varphi_c = \varphi_v = \theta/2$ (a). When the camera has an offset $L$ with the vehicle's origin, we still have $\varphi_v = \theta/2$ but $\varphi_c \neq \theta/2$ (b). Note, the camera does not necessarily have to be on the axis of symmetry of the vehicle.

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.

# 4. Absolute scale computation

## 4.1. Camera undergoing planar circular motion

Figure 3(b) shows the camera and vehicle coordinate systems. Both coordinate systems are aligned so that there is no additional rotation between them. Observe that the camera does not necessarily have to be on the axis of symmetry of the vehicle. The camera is denoted by $P_1$ and it is located at $C_1 = [0, 0, L]$ in the vehicle coordinate system. The camera matrix $P_1$ is therefore

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -L \end{bmatrix} \tag{1}$$

The camera $P_1$ and the vehicle now undergo the following circular motion denoted by the rotation $R$ and the translation $T$ (see also Fig. 3(b)).

$$R = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{2}$$

$$T = \rho \begin{bmatrix} \sin\left(\frac{\theta}{2}\right) \\ 0 \\ \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{3}$$

The transformed camera $P_2$ is then

$$P_2 = [R_2 \;\; t_2] = P_1 \begin{bmatrix} R & -RT \\ \mathbf{0} & 1 \end{bmatrix} \tag{4}$$

To compute the motion between the two cameras $P_2$ and $P_1$, the camera $P_2$ can be expressed in the coordinate system of $P_1$. Let us denote it by $P'_2$.

$$P'_2 = [R'_2 \;\; t'_2] = P_2 \begin{bmatrix} & P_1 & & \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \tag{5}$$

The rotation part $R'_2$ equals $R_2$ (which equals $R$) and the translation part $t'_2$ is

$$t'_2 = \begin{bmatrix} \rho\sin\left(\frac{\theta}{2}\right) - L\sin(\theta) \\ 0 \\ L\cos(\theta) - \rho\cos\left(\frac{\theta}{2}\right) - L \end{bmatrix} \tag{6}$$

Then, the essential matrix $E$ for our setup describing the relative motion from camera $P_1$ to $P'_2$ is defined as $E = [t'_2]_\times R'_2$ and can be written as:

$$E = \begin{bmatrix} 0 & L + \rho\cos\left(\frac{\theta}{2}\right) - L\cos(\theta) & 0 \\ L - \rho\cos\left(\frac{\theta}{2}\right) - L\cos(\theta) & 0 & \rho\sin\left(\frac{\theta}{2}\right) + L\sin(\theta) \\ 0 & \rho\sin\left(\frac{\theta}{2}\right) - L\sin(\theta) & 0 \end{bmatrix} \tag{7}$$

Finally, observe that the translation part $t'_2$ and the essential matrix $E$ can also be expressed in terms of the absolute distance $\lambda$, between the two camera centers, and the camera relative motion $(\theta, \varphi_c)$. Thus, we obtain:

$$t'_2 = \lambda \begin{bmatrix} \sin(\theta - \varphi_c) \\ 0 \\ -\cos(\theta - \varphi_c) \end{bmatrix} \tag{8}$$

$$E = \lambda \begin{bmatrix} 0 & \cos(\theta - \varphi_c) & 0 \\ -\cos(\varphi_c) & 0 & \sin(\varphi_c) \\ 0 & \sin(\theta - \varphi_c) & 0 \end{bmatrix} \tag{9}$$

where the latter is the standard expression of the essential matrix for general planar motion. These two expressions for $t'_2$ and $E$ will be used in the next sections for computing the absolute scale.

### 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 $\lambda = ||t'_2||$ (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 $\varphi_c$ of the camera translation vector because these parameters can be estimated from feature correspondences. For this we equate the two expressions for $t'_2$ (6) and (8). We obtain the following two equations:

$$\begin{array}{rcl} \rho\sin\left(\frac{\theta}{2}\right) - L\sin(\theta) & = & \lambda\sin(\theta-\varphi_c) \\ L\cos(\theta) - \rho\cos\left(\frac{\theta}{2}\right) - L & = & -\lambda\cos(\theta-\varphi_c) \end{array} . \qquad (10)$$

From this we can get the expressions for $\rho$ and $\lambda$ in terms of $\theta$ and $\varphi_c$.

$$\rho = \frac{L\sin(\varphi_c) - L\sin(\varphi_c-\theta)}{\sin(\varphi_c - \frac{\theta}{2})} \qquad (11)$$

$$\lambda = \frac{2L\sin(\frac{\theta}{2})}{\sin(\varphi_c - \frac{\theta}{2})} \qquad (12)$$

Note, expressions (11) and (12) are exactly the core of this paper, that is, we can actually compute the absolute distance between the vehicle or the camera centers as a function of the camera offset $L$ and the camera relative motion $(\theta, \varphi_c)$. In the next section we will give a minimal and a least-square solution to compute $\theta$ and $\varphi_c$ directly from a set of point correspondences. Finally, note that $\rho$ and $\lambda$ are valid only if $L \neq 0$ and $\theta \neq 0$. Thus, we can only estimate the absolute scale if the camera has an offset to the vehicle center of motion and when the vehicle is turning. Note also that in order to have $\rho > 0$ and $\lambda > 0$, we must have $\varphi_c > \theta/2$ if $\theta > 0$ or $\varphi_c < \theta/2$ if $\theta < 0$. The other way round, if $L < 0$. 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 (11) with respect to $L$ we have:

$$L = \rho\frac{-\sin(\frac{\theta}{2} - \varphi_c)}{\sin(\varphi_c) + \sin(\theta - \varphi_c)} \qquad (13)$$

## 5. Planar motion estimation

In the previous section, we used the circular motion constraint to compute the absolute scale from generic $\theta$, $\varphi_c$. Here, we describe two algorithms to estimate up to scale motion (i.e. $\theta$, $\varphi_c$) of a calibrated camera under planar assumption. We provide a least-square solution, which requires a minimum of 3 point correspondences, as a well as a minimal solution which only needs two point correspondences. These algorithms are valid for both perspective and omnidirectional cameras. However, to avoid bad data conditioning we recommend to normalize all image points on the unit sphere.

### 5.1. 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

$$p'^T E p = 0 \qquad (14)$$

Using the expression (9) of the essential matrix, the epipolar constraint expands to:

$$\begin{array}{ccccc} -xy'\cos(\varphi_c) & + & yx'\cos(\theta-\varphi_c) & + & \\ zy'\sin(\varphi_c) & + & yz'\sin(\theta-\varphi_c) & = & 0. \end{array} \qquad (15)$$

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

$$\begin{array}{rclrcl} h_1 & = & -\cos(\varphi_c), & h_2 & = & \cos(\theta-\varphi_c) \\ h_3 & = & \sin(\varphi_c), & h_4 & = & \sin(\theta-\varphi_c). \end{array} \qquad (16)$$

Note, as the solution is valid up to a scale, we actually need at least 3 point correspondences to find a solution.

Finally, the angles $\theta$ and $\varphi_c$ can be derived by means of a four-quadrant inverse tangent. However, as the elements of the unknown vector are not independent of each other, nonlinear optimization could be applied to recover more accurate estimations. The next section covers how to deal with it.

### 5.2. Minimal solution: non-linear 2-point algorithm

This method proceeds along the lines of Ortin and Montiel [9]. As shown by Eq. (15), 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,

$$f(\theta,\varphi_c) \approx f(\theta_0,\varphi_{c0}) + J_f(\theta_0,\varphi_{c0}) \begin{bmatrix} (\theta-\theta_0) \\ (\varphi-\varphi_{c0}) \end{bmatrix} \qquad (17)$$

where $f(\theta_0, \varphi_{c0})$ can be computed from (15) and the Jacobian $J_f(\theta_0, \varphi_{c0})$ can be written as:

$$J_f(\theta_0,\varphi_{c0}) =$$

$$\begin{bmatrix} -yx'\sin(\theta_0-\varphi_{c0}) + yz'\cos(\theta_0-\varphi_{c0}) \\ xy'\sin(\varphi_{c0}) + yx'\sin(\theta_0-\varphi_{c0}) - yz'\cos(\theta_0-\varphi_{c0}) + zy'\cos(\varphi_{c0}) \end{bmatrix} \qquad (18)$$

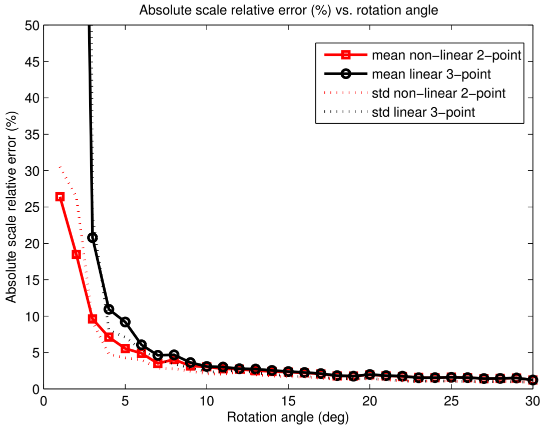


Figure 4. The relative error $\%$ of the absolute scale estimate as a function of the rotation angle $\theta$. Comparison between the linear 3-point method (circles) and the non-linear 2-point method (squares).

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:

$$\begin{bmatrix} \theta_{i+1} \\ \varphi_{ci+1} \end{bmatrix} = J_f(\theta_i, \varphi_{ci})^{-1} f(\theta_i, \varphi_{ci}) + \begin{bmatrix} \theta_i \\ \varphi_{ci} \end{bmatrix} \tag{19}$$

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. In the final implementation, section 6.2, we used the 2-point algorithm because of its slightly better performance.

# 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 shown in equation (11), the estimate of the absolute scale $\rho$ from the camera relative motion is only possible 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 and a digital color camera ( image size $640 \times 480$ pixels). The camera was installed as shown in Fig. 1(b). The offset of the camera from the rear axle is $L$=0.9m. The camera system was calibrated using the toolbox from Scaramuzza [13, 11]. 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 under planar constraint was performed using the 2-point algorithm of section 5.2. We did this for the all 4000 frames of the dataset. In addition to the visual measurements, we also had 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.

### 6.2.1 Circular motion detection

The equations (11) and (12) for absolute scale estimation only give 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 perfectly circular

- Algorithm:
  - Compute camera motion estimate up to scale
  - Compute absolute scale ($\rho$) from $\theta, \varphi_c, L$
  - Identify sections for which $\rho > 0$
  - Identify sections for which the circular motion is satisfied
    - Compute curvatures of two neighboring sections: $k_i$, $k_{i+1}$
    - Check circular motion criterion: $\frac{|k_i - k_{i+1}|}{k_i} < 10\%$
  - Consider correct absolute scale for sections for which $|\theta| > \theta_{thresh}$

Figure 5. Outline of the absolute scale algorithm

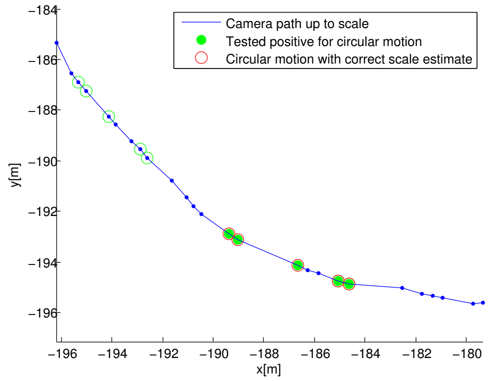


Figure 6. Section of the camera path that shows circular motion. Red and green circles mark respectively beginning and end of circular motion section from which a correct absolute scale was computed.

motion, the curvature $k$ is constant. The idea is therefore to look for motion that satisfies this condition. In practice, we accept relative changes up to $10\%$. Furthermore, in order to exclude paths with low curvature, we only consider sections with curvature values between $0.03$ and $0.5\ m^{-1}$ and, therefore, with radius between 2 and 33 meters.

The curvature of a circle of radius $r$ is defined as $k = 1/r$. The radius can be trivially computed using trigonometry by observing that an arc of length $r\theta$ has chord length $c = 2r\sin(\theta/2)$. In our case, the chord length is exactly the distance $\rho$ which is computed from (11). Therefore, we can set $c = \rho$ and the curvature $k$ can be then readily obtained as:

$$k = \frac{2\sin(\frac{\theta}{2})}{\rho} \tag{20}$$

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 $i$, we compute $k_i$ using (20) and we check if it represents circular motion by checking the difference with the curvature $k_{i+1}$ of the next frame pair. If this difference is smaller than $10\%$, the motion is classified as circular and non-circular otherwise. Observe that prior to computing the curvature, we check if $\rho > 0$. If this condition is not satisfied, the section is excluded a priori. The basic outline of the algorithm is described in Fig. 5.

### 6.2.2 Results on absolute scale estimation

Fig. 6 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 circular motion are indicated by green dots. The section starts at the red circles and ends at the green circles. The sections were detected as described above. For each detected circular 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.

Results on the accuracy of the method and on the influence of the threshold on $\theta$ are shown in Table 1 for the whole 3 Km path. Here we list the number of all detected circular motion sections and the number of correctly computed scales. By tuning the threshold $\theta_{thresh}$, the results can be optimized. The absolute scale of such a section will only be computed if the turning angle $\theta$ is larger than $\theta_{thresh}$. With a larger threshold on $\theta$ it is possible to reduce the of number wrong computed scales. With a threshold setting of $30°$ it was possible to remove all wrong estimates. With this setting, eight 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.

Fig. 7 shows a plot of the full path. Sections satisfying $\rho > 0$ are shown as green dots. Observe that more than $50\%$ of the sections satisfy this condition. Sections where motion was classified as circular are shown as red circles. Circular motion appears not only in sharp turns but also at slight curves. Finally, blue 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.

| $\theta_{thresh}[^\circ]$ | # detected | # correct |
|---|---|---|
| 5 | 461 | 193 |
| 10 | 153 | 65 |
| 20 | 36 | 21 |
| 30 | 8 | 8 |

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

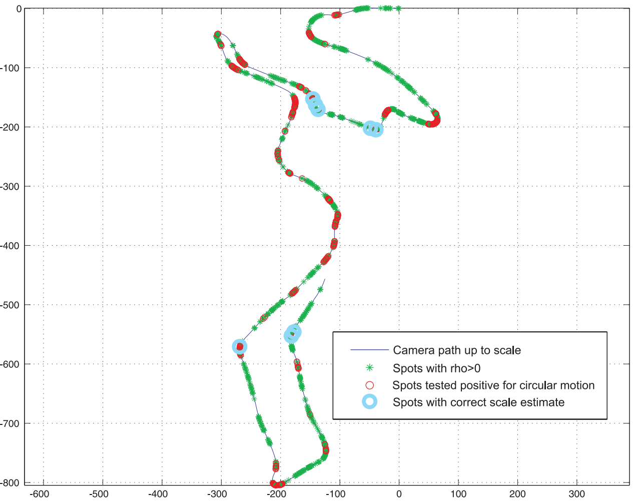


Figure 7. Camera path (3 Km) showing sections of circular motion (red circles) and sections with correct estimated absolute scale (large blue circles).

## 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 circular motion.

Our experiments show that a good accuracy can be achieved even with a small offset like 0.9m, although a larger offset would increase the accuracy. The camera does not need to be placed at a specific place, nor on the axis of symmetry of the vehicle. 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

[1] B. Clipp, J. Kim, J. Frahm, M. Pollefeys, and R. Hartley. Robust 6dof motion estimation for non-overlapping, multi-camera systems. In *Proc. IEEE Workshop on Applications of Computer Vision*, pages 1–8, 2008.

[2] A. Davison. Real-time simultaneous localisation and mapping with a single camera. In *International Conference on Computer Vision*, 2003.

[3] G. Jiang, L. Quan, and H. Tsui. Circular motion geometry using minimal data. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2004.

[4] G. Jiang, H. Tsui, L. Quan, and A. Zisserman. Single axis geometry by fitting conics. In *ECCV*, 2002.

[5] I. Jung and S. Lacroix. Simultaneous localization and mapping with stereovision. In *Robotics Research: the 11th International Symposium*, 2005.

[6] M. Maimone, Y. Cheng, and L. Matthies. Two years of visual odometry on the mars exploration rovers: Field reports. *Journal of Field Robotics*, 24(3):169–186, 2007.

[7] H. Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. In *tech. report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University*. September 1980.

[8] D. Nister, O. Naroditsky, , and B. J. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 2006.

[9] D. Ortin and J. Montiel. Indoor robot motion based on monocular images. *Robotica, vol. 19, issue 3, p.331-342, May 2001*.

[10] M. Pollefeys, D. Nister, J. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, R. Y. H. Stewenius, G. Welch, and H. Towles. Detailed real-time urban 3d reconstruction from video. *International Journal of Computer Vision*.

[11] D. Scaramuzza. Ocamcalib toolbox: Omnidirectional camera calibration toolbox for matlab, 2006. Google for ”*ocamcalib*”.

[12] D. Scaramuzza, F. Fraundorfer, and R. Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In *IEEE International Conference on Robotics and Automation (ICRA 2009), Kobe, Japan, 16 May*, 2009.

[13] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easy calibrating omnidirectional cameras. In *IEEE International Conference on Intelligent Robots and Systems (IROS 2006)*, oct 2006.

[14] D. Scaramuzza and R. Siegwart. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics*, 24(5), October 2008.

[15] R. Siegwart and I. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. MIT Press, 2004.