Proceedings of the 2007 IEEE
Intelligent Transportation Systems Conference
Seattle, WA, USA, Sept. 30 - Oct. 3, 2007

Extrinsic Camera Calibration for an On-board Two-Camera System without overlapping Field of View

Bernhard Lamprecht and Stefan Rass and Simone Fuchs and Kyandoghere Kyamakya
Institute for Smart-System Technologies
Klagenfurt University
9020 Klagenfurt, Austria
{firstname.lastname}@uni-klu.ac.at

Abstract— **Most recent developments in car technology promise that future cars will be equipped with many cameras facing different directions (e.g.: headlights, wing mirrors, break lights etc.). This work investigates the possibility of letting the cameras calibrate and localize themselves relative to each other by tracking one arbitrary and fixed calibration object (e.g.: a traffic sign). Since the fields of view for each camera may not be overlapping, the calibration object serves as logical connection between different views. Under the assumption that the intrinsic camera parameters and the vehicle's speed are known, we suggest a method for computing the extrinsic camera parameters (rotation, translation) for a two-camera system, where one camera is defined as the origin.**

I. INTRODUCTION

Recent developments in the car-industry lead to the assumption that future cars will be equipped with cameras observing different areas around the car. It appears likely that in future, cameras will be integrated into cars' headlights, wing mirrors, break lights etc. Nowadays technology allows a shapely and easy integration of sensors into cars, however, cameras as well as many other parts of the car, may be subject to replacement due to age, damage or technological improvement. Although automated mounting during the production process may be capable of determining the camera's rotation and position accurately, a human engineer replacing the camera will most likely be unable to determine these parameters exactly again. Self-calibrating cameras spare difficult and time-consuming manual measurements for calibration. If the camera is replaced by a different model or if the position is changed for any other reason, a re-calibration may become necessary. Even without significant changes in the car's geometry, a periodic re-calibration is advisable.

Current vision-enabled driver assistance systems mostly operate in one direction. A lane-departure warning system [10] senses the environment in driving direction, a blind-spot warning system [3] observes the side area of the car. This work investigates the idea of fusing different camera views to one comprehensive environment view. It turns out that such an approach requires a fully calibrated camera system. A multi-camera system (cf. figure 1) attached to a car is completely determined by the *intrinsic parameters* plus position and rotation (subsumed as *extrinsic parameters*) of each camera.

We are developing a system for auto-calibration of a multi-camera system for a car-like robot. The robot is equipped with cameras facing different directions and serves as testbed. In this paper we present the mathematical framework for calibrating a two-camera system where one camera is defined as the origin. Due to the field of application the cameras have disjoint fields of views (FoV). We assume the camera positions to be determined by car-design issues, which means they are arbitrary. The camera calibration process can be performed on-line while driving straight across a street with constant speed. An arbitrary but fixed point is used as calibration point. Such a point can be a traffic sign for instance. Figure 1 shows the principle for on-line auto-calibration. CAM1 and CAM2 recognize and identify a traffic sign. The traffic sign is tracked while it is in FoV of the according camera. In figure 1, the traffic sign will appear first in the FoV of CAM1, and then will be seen by CAM2. During the tracking process projection equations are generated. As soon as the tracking point has left the FoV of CAM2, the solution of the resulting system of non-linear equations provides CAM2's location and rotation.

Our proposal for a fully auto-calibration of a moving multi-camera system consists of three independent functional modules:

1) Tracking of traffic signs between cameras with non overlapping FoV.
2) Calculation of intrinsic camera parameters.
3) Calculation of extrinsic camera parameters.

This paper covers the third module. We assume the intrinsic parameters to be known a priori or that they can be determined automatically during driving and that there exists an algorithm to track and recognize a traffic sign between different camera views with non overlapping FoV [9].

The next section discusses related work. Section IV is dedicated to a detailed description of the system equations and the algorithm for self-calibration. We analyze the system and corresponding simulation results regarding measurement error sensitivity in section V. The last section summarizes this work and gives an overview of our future research in this field.

1-4244-1396-6/07/$25.00 ©2007 IEEE.

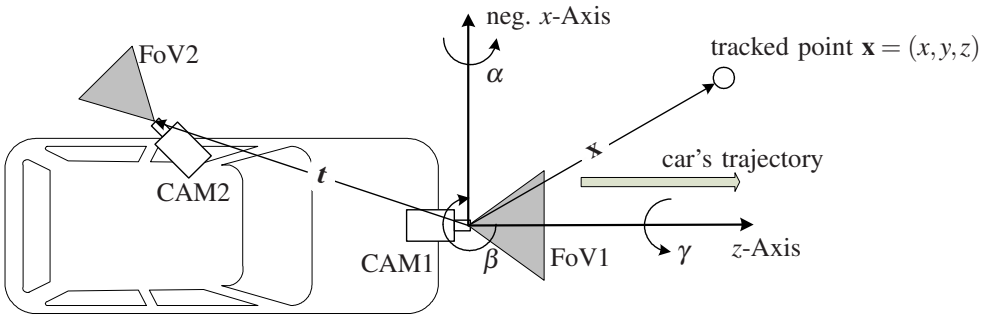


Fig. 1. Principle of Self-Calibration. The vector $t = (t_x^2, t_y^2, t_z^2)$ is the position of CAM2 relative to the origin where CAM1 is located. The y-axis is viewed from negative infinity, i.e. the positive y-direction goes "into" the picture plane. Positive angles represent anti-clockwise rotation. The rotation of CAM1 is assumed zero and the origin is located at CAM1.

II. RELATED WORK

The extrinsic calibration of multi-camera systems has attracted considerable amount of attention. One distinguishing feature for all existing approaches is the FoV for each camera - *overlapping vs. non-overlapping*. In the overlapping case [5][2][12][6] every camera of the camera rig has at least a common FoV with a second camera of the system. This common region is used to establish the logical connection between the cameras for the calibration process. In the latter case [8][11][1][4] no or not all cameras of the system share a common FoV, as in our approach (cf. figure 1).

A. overlapping FoV

A fully distributed approach for calibration of a camera network is presented in [6]. The cameras collaborate to track an object that moves through the environment and reason probabilistically about which camera poses are consistent with the observed images. In [2] a calibration procedure for a multi-camera system is proposed. The cameras calibrate itself based on point correspondences. The correspondences are then used in a large, nonlinear eigenvalue minimization routine. For a moving multi-camera system, in [5] a initial calibration technique is presented to estimate the rotational and translational component for each camera of the multi-camera system. The initial calibration has to be done once to determine the configuration of the multi-camera system.

B. non-overlapping FoV

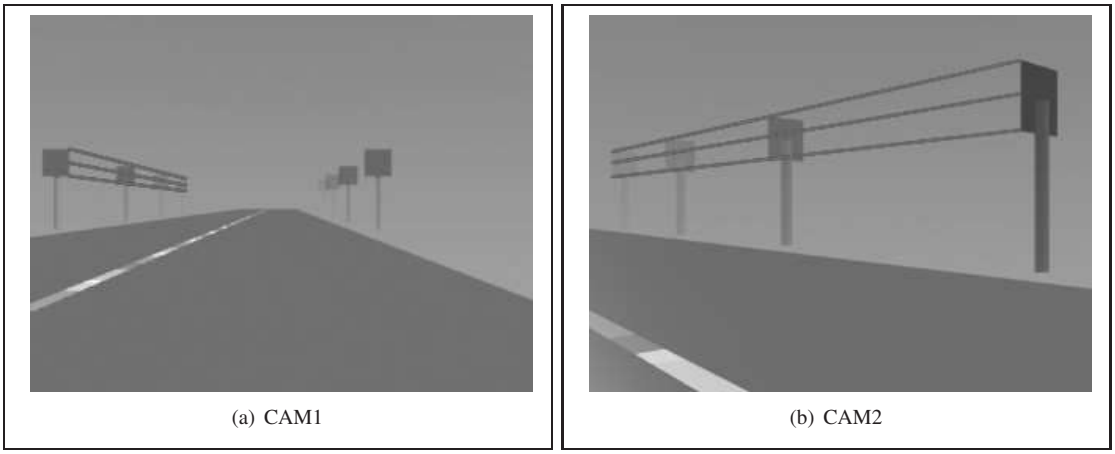
In the non-overlapping case there is the need for the compensation of the lack of overlap between the cameras' FoV. The approach of Junejo et al. [8] shows that only one automatically computed vanishing point and a line lying on any plane orthogonal to the vertical direction is sufficient to infer the cameras rotation. They state that it is not possible to estimate the relative translation between two disjoint FoV cameras. Our approach differs in that way that we can compute next to the rotation also the translation by incorporating the car's speed. Rahimi et al. [11] describe a method for recovering the external calibration parameters of non-overlapping cameras in a multi-camera system by tracking a moving target (e.g.: human). The lack of overlap is compensated by prior knowledge of the target's dynamics. The target is allowed to move freely and with varying speed and direction. A similar approach with a different motion model of the calibration target can be found in [1]. The work of [11] and [1] use probability distributions to model the trajectory of the calibration object. The computation of the final external camera parameters is an iterative process. The more calibration objects are observed the better the estimate of rotation and translation of each camera will be. This paper proposes an approach where only one calibration object passes by only once at each camera and yield to an immediate solution. Furthermore we assume that the calibration object follows a straight trajectory due to the straight movement of the car. This paper is most closely related to the work of Fisher [4]. In his work he suggests a method to calibrate a set of randomly placed sensors. He uses distant moving objects (the motion of stars is parabolic) for rotation determination and nearby linearly moving features (airplanes with constant speed) for full pose registration. Common to our work is that he assumes the trajectory of the calibration object to be known. He uses two different kinds of calibration objects (airplanes and stars), we use only one (traffic signs).

C. Contributions of this work

We suggest a simple method for computing the extrinsic camera parameters for an on-board two-camera system without overlapping FoV. The proposed solution requires only one calibration object (e.g.: traffic sign) that passes by once during straight driving with constant speed.

III. PRELIMINARIES AND NOTATION

We denote points in the world as column vectors $\mathbf{x} \in \mathbb{R}^3$, and points in the picture plane by $\mathbf{u}$. Lower-case bold-face characters are always vectors, while upper-case bold-face letters denote matrices. Anything not bold printed is a scalar. Subscripts in greek letters (τ) refer to the time of sampling, superscripts associate a value with a camera. A subscript in latin letters (t_x for instance), refers to the x-coordinate of the vector $\mathbf{t}$. We assume the reader to be familiar with projective geometry in computer vision.



(a) CAM1 (b) CAM2

Fig. 2. Calibration points that are obtained by the observation of a traffic sign in CAM1 and CAM2. For representation the calibration points are connected with a line. Here, one traffic sign provides three calibration points (up, middle, bottom). In the left subfigure (a) the left and most far away image of the traffic sign is observed first during the calibration process ($\tau = 0$).

IV. MODEL DERIVATION

This section describes the derivation of the model that provides estimates for the *extrinsic camera parameters*, as well as the calibration point coordinates (as a by-product).

A. Projection Model

A point $\mathbf{x}$ with world coordinates (x,y,z) (relative to CAM1; cf. figure 1) is subject to a set of transformations when being seen by CAM1:

- Movement and rotation by the camera's position and location (determined by the extrinsic parameters),
- Projection onto the image plane and discretization into pixels (determined by the intrinsic parameters), and
- translation relative to the car's position (as the car moves).

For convenience, we represent column-vectors $(x,y,z)^T \in \mathbb{R}^3$ by adding a fourth coordinate being constant 1. The representation $\mathbf{x} = (x,y,z,1)^T$, allows for doing movements and rotations by a single matrix multiplication. We thus model the aforementioned transformations by matrix multiplications in the above order: At time τ, let $\mathbf{u}^i_\tau \in \mathbb{R}^3$ be the point visible in the image of the i-th camera, and let $\mathbf{x} \in \mathbb{R}^4$ be the tracked point (traffic sign). Note that although $\mathbf{u}$ is a point in a plane image, the model provides us with the depth information as an unknown parameter (as this information is lost by the projection on a lower dimensional space). We will remove this coordinate later by substitution. But first, let us give the projection equation. The superscript index $i \in \{1,2\}$ is used to distinguish between the parameters of CAM1 and CAM2:

$$\mathbf{u}^i_\tau = \mathbf{I}^i \mathbf{E}^i \mathbf{M}_\tau \mathbf{x}, \tag{1}$$

where the matrices $\mathbf{I}, \mathbf{E}$ and $\mathbf{M}_\tau$ are defined as follows:

$$\mathbf{I}^i = \begin{pmatrix} f^i_u & 0 & m^i_0 & 0 \\ 0 & f^i_v & n^i_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

is the intrinsic parameter matrix with f_u, f_v being the focal lengths and m_0, n_0 determining the principal point of the camera. Secondly, the extrinsic camera parameters are defined as block-matrices of the form

$$\mathbf{E}^i = \left(\begin{array}{c|c} \mathbf{R}^i & \mathbf{t}^i \\ \hline 0 & 1 \end{array}\right) \in \mathbb{R}^{4\times 4},$$

where $\mathbf{R}$ is the camera's rotation. We treat these as a black-box in our model. The vector $\mathbf{t}^i = (t^i_x, t^i_y, t^i_z)$ gives the position of CAM2 relative to CAM1.

A motion of the car along the z-axis (cf. figure 1) is encoded by a simple matrix of the form

$$\mathbf{M}_\tau = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_\tau \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where d_τ is the distance traveled at time τ. τ is set to zero when the traffic sign is recognized first in CAM1.

From figure 1, it is evident that CAM1 is the origin, hence its extrinsic parameter $\mathbf{t}^1$ is zero. We shall derive an equation system for finding the parameter $\mathbf{t}^2$. The output of the transformation (1) is of the form:

$$\mathbf{u}^i_\tau = (u^i_\tau w^i_\tau, v^i_\tau w^i_\tau, w^i_\tau)^T, \tag{2}$$

The first two variables u^i_τ and v^i_τ are the pixel coordinates. The third, yet unknown, value w^i_τ appears as coefficient of the first two variables. Dividing by this value gives two equations (the third is of the form "$1 = 1$" and can thus be dropped, such as the sub- and superscripts are omitted for simplicity and generality):

$$\frac{1}{w}\mathbf{I}\cdot\mathbf{E}\cdot\mathbf{M}\cdot\mathbf{x} - \mathbf{u} = 0. \tag{3}$$

The unknown values in equation (3) are contained in $\mathbf{E}$ and $\mathbf{x}$. Carrying out the matrix multiplication uncovers that this is a nonlinear system of two equations for a single point.

It should be mentioned that the projection is indeed

non-unique [7], due to an infinite number of equivalent rotation angles and the fact that a point in front of the camera will be mapped to the same position as a (theoretical) point behind the camera. Although physically impossible, we cannot implement this in the model. However, as figure 2 shows, this situation can be ruled out by our camera setup.

B. Concept

Let us initially assume that the camera rotations are known. We use the projection equation (1) as a starting point, and assume the car to be equipped with two cameras as depicted in figure 1. The on-board vision system is capable of tracking an arbitrary but fixed point $\mathbf{x}$ across the FoV of both cameras. The idea is to track $\mathbf{x}$ twice (at times τ_1, τ_2) while it is inside the FoV of CAM1. This gives the points $\mathbf{u}^1_{\tau_1}, \mathbf{u}^1_{\tau_2}$. Equation (1) then provides 6 linear constraints on 5 unknowns, $x, y, z, w^1_{\tau_1}, w^1_{\tau_2}$, two of which ($w^1_{\tau_1}$ and $w^1_{\tau_2}$), in turn can be substituted thanks to equation (2)). This leaves us with 4 equations with 3 unknowns x, y, z.

Repeating the procedure as soon as $\mathbf{x}$ enters the FoV of CAM2, we track $\mathbf{x}$ at times τ_3, τ_4 and substitute the third component of $\mathbf{u}^2_{\tau_3}, \mathbf{u}^2_{\tau_4}$ to get another 4 equations, now with 6 unknowns x, y, z and t^2_x, t^2_y, t^2_z (these give the position relative to CAM1; consequently, for CAM1, $t^1_x = t^1_y = t^1_z = 0$).

Theoretically, this over-determined system could be solved using a standard least-squares approximation. Simulation showed that the accuracy of the output is strongly dependent on the accuracy of the rotation parameters: We conducted an experiment (using the setup we describe in section V) where we added a random noise to the error. It turned out that even a deviation by 1 degree causes an error of > 0.8 length units. Taking into account that the coordinates for our experimental setup are given in meters, this yields an error of 80 cm, which is unacceptable.

Hence, we generalize the model to letting the camera rotation be an unknown parameter. Furthermore this saves us from lengthy and expensive analysis in order to estimate the cameras rotations. Instead of tracking one point per camera, we track *three* points *five times per camera* (cf. figure 2), and solve the resulting system of $(2 \times 3 \times 5 = 30)$ nonlinear equations (arising from $3 \times 5 = 15$ instances of equation (3)) numerically, using a least-squares approximation. A presentation of the results is subject of the next section.

V. EXPERIMENTS

We conducted an experiment with the following configuration: The cameras are located as shown in figure 1, with distance vector $\mathbf{t} = (-1, -0.2, -1)^T$ between CAM1 and CAM2 (all coordinates are relative to the origin shown in figure 1). A meter is used as length-unit. The FoV of both cameras is assumed 60° wide. The rotation of CAM1 is $(0,0,0)$ (i.e. the FoV of CAM1 is centered along the car's trajectory), and the rotation of CAM2 is $-140°$ around the y-axis and 0° around the x, y-axes. (cf. figure 1). The choice of angles is such that the resulting FoV covers the dead sector (w.r.t. the driver's perspective under the assumption that s/he can turn the head up to an angle of 20° around the y-axis shown in figure 1).

It appears reasonable to take traffic signs as tracking points. For that matter, in accordance with the coordinate axis configuration shown in figure 1, we can assume that the tracked points are located 5 meters to the side of the car and ≈ 2 meters above the ground. With an assumed frame rate of $25fps$ (a standard value), a traffic sign being 8 meters ahead of the car, at a speed of $30km/h$, the calibration point leaves the FoV of CAM1 after $\approx 0.54s$, i.e. we can take at most 13 samples (i.e. CAM1 provides 13 frames). For CAM2, a direct calculation, with the given configuration of angles and velocity, shows that the traffic sign will visible in the FoV of CAM2 for a duration of $\approx 1.27s$, which gives at most 31 frames, i.e. a maximum of 31 samples. In total we get at most $\min\{31, 13\} = 13$ samples. We thus increase the number of samples for CAM1 by assuming the point at 20 m distance initially.

According to the idea presented in section IV-B and the previous paragraphs, we tracked three points with coordinates $\mathbf{x}_{P1} = (-5, -1.75, 20), \mathbf{x}_{P2} = (-5, -2, 20)$ and $\mathbf{x}_{P3} = (-5, -2.25, 20)$ five times for each camera (the traffic sign is 0.5 m in height), and solved the resulting nonlinear system (cf. figure 2). The initial guess is provided by the distances and rotations that have been determined at the first-time mounting of the cameras by the car manufacturer. It is reasonable to assume that the cameras are mounted by robots and thus the positions are known quite exactly. However, a human engineer changing or repairing a camera will tend to install it almost as it has been previously, but won't achieve the same accuracy. Small differences in the setting will be compensated by the auto-calibration, but the manufacturer's settings may provide a sufficiently accurate initial guess.

For our simulation, we initiated 10 trials. In each trial, we randomly altered the system parameters (calibration points, velocity and tracking precision): the calibration points were chosen randomly within a neighborhood of the specified points $\mathbf{x}_{P1..3}$. Pixel roundoff errors are implemented by rounding the projected coordinates. Normally distributed random errors for tracking and the speed were added for a realistic simulation. Assuming that we travel a total distance of 50 m, errors in the velocity measurement were assumed to be at most 1 m over this distance[1], and the parameter of the normal distribution were (discretely) varied from 0 (exact measurement) up to the full error of ± 1 m over 50 m

[1]GPS navigation systems by GARMIN (see `http://www.garmin.de/`) achieve an accuracy of about 0.05 m/s when measuring velocity, which results in at most ± 0.3 m over a distance of 50 m (at 30 km/h).

distance. The tracking error is (discretely) varied from 0 (exact tracking) up to ± 10 pixels.

The intrinsic camera parameters used for the simulation are $f_u^1 = 555 = f_u^2 = f_v^1 = f_v^2$. The resolution is 640×480, i.e. $m_0^1 = m_0^2 = 320, n_{01} = n_0^2 = 240$. We assumed no distortion and the intrinsic camera parameters to be constant over all simulations.

Figure 3 displays the precision of the position determination depending on the given accuracies for the speed and tracking. The z-values are the mean euclidian distance errors over all trials with given accuracies. As expected, the location of CAM2 relative to CAM1 is determined quite precisely for small distance errors. However, for a realistic accuracy, the results become unacceptably wrong at a speed of 50 km/h. Interestingly, the tracking error seems to have rather small influence on the precision, as the diagram shows.

Figure 4 shows the precision of the angle estimation under the same setup as figure 3, but displays the *maximum* deviation of angles from their true values (i.e. the distance w.r.t. the $\|\cdot\|_\infty$-norm). Intuitively, one would expect the angles α and β to be computable quite precisely, while the rotation γ (cf. fig. 1) is not precisely calculable from the given 1-dimensional motion of the vehicle. And in fact, this is the main reason for the sometimes large deviation displayed in figure 4. Figure 5 shows the deviations of each angle separately depending on the tracking error, but with fixed accuracy of speed measurement. As mentioned before (see footnote 1), an accuracy of 0.05 m/s is realistic. In this case, figure 5 shows that α, β are estimated with good precision, while γ significantly deviates, which explains the shape of the surface in figure 4.

VI. CONCLUSIONS AND FUTURE WORK

The results of our work show that it seems indeed possible to have the cameras self-calibrate during driving along a straight line by tracking an object across the field of view of a single camera. With the assumption of non-overlapping fields of views for two cameras, we can determine the cameras extrinsic parameters just by moving along a straight line for a few seconds.

While the system is yet unable to determine the third angle and the translation with an accurate precision, we believe that this is also possible by simple modifications. Future work includes the incorporation of an inertial system into the calibration process to relax the assumption of straight movement and constant speed. The results are promising, as the system is easy to implement, computationally cheap, and does not rely on any sophisticated hardware, except for a GPS receiver / inertial system, which we believe to become standard in future vehicles that are equipped with a multi-camera system.

REFERENCES

[1] N. Anjum, M. Taj, and A. Cavallaro. Relative position estimation of non-overlapping cameras. In *ICASSP: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Honolulu, USA, April, 2007.

[2] Patrick Baker and Yiannis Aloimonos. Complete calibration of a multi-camera network. *omnivis*, 00:134, 2000.

[3] D.S. Breed. Vehicular blind spot identification and monitoring system. United States Patent 7049945, May, 23rd 2006. `http://www.freepatentsonline.com/7049945.html`.

[4] R. Fisher. Self-organization of randomly placed sensors, 2002.

[5] J.M. Frahm, K. Koser, and R. Koch. Pose estimation for multi-camera systems. In *DAGM04*, pages 286–293, 2004.

[6] Stanislav Funiak, Carlos Guestrin, Mark Paskin, and Rahul Sukthankar. Distributed localization of networked cameras. In *IPSN '06: Proceedings of the fifth international conference on Information processing in sensor networks*, pages 34–42, New York, NY, USA, 2006. ACM Press.

[7] Hartley and Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[8] Imran Junejo, Xiaochun Cao, and Hassan Foroosh. Geometry of a non-overlapping multi-camera network. In *AVSS '06: Proceedings of the IEEE International Conference on Video and Signal Based Surveillance*, page 43, Washington, DC, USA, 2006. IEEE Computer Society.

[9] Dimitrios Makris, Tim Ellis, and James Black. Bridging the gaps between cameras. *cvpr*, 02:205–210, 2004.

[10] T. Pilutti and A.G. Ulsoy. Fuzzy-logic-based virtual rumble strip for road departure warning systems. *IEEE Transactions on Intelligent Transportation Systems*, 4, Issue 1:1–12, 2003.

[11] A. Rahimi, B. Dunagan, and T. Darrell. Simultaneous calibration and tracking with a network of non-overlapping sensors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–187–I–194Vol.1, 27 June-2 July 2004.

[12] Richard G. Baraniuk William E. Mantzel, Hyeokho Choi. Distributed camera network localization. 2004.

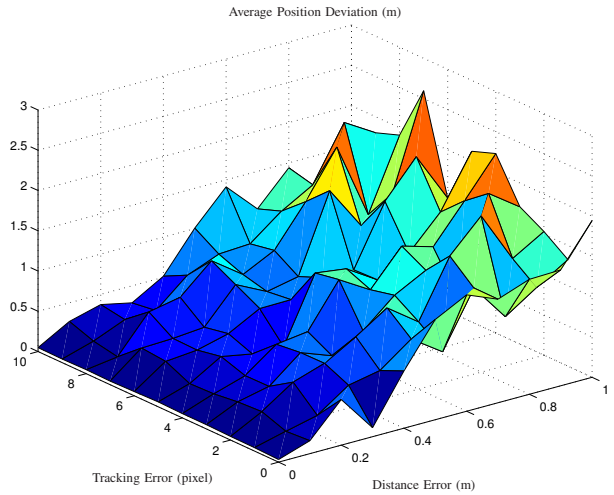


Fig. 3. Location error dependent on tracking and distance(speed) errors.

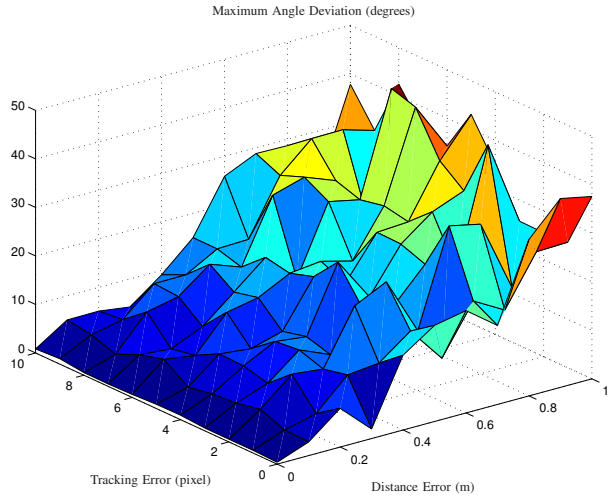


Fig. 4. Angle deviation dependent on tracking and distance(speed) errors.

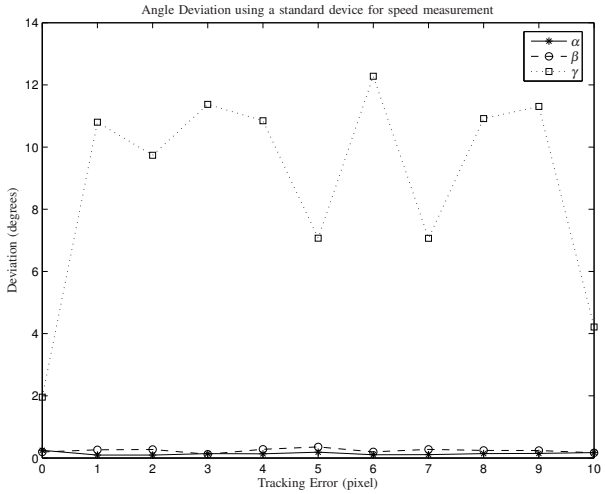


Fig. 5. Accuracy of angles α, β and γ.