Robust Time-to-Contact Calculation for Real Time Applications

Maria Sagrebin, Anastasia Noglik, Josef Pauli
Fakultät für Ingenieurwissenschaften,
Abteilung für Informatik und Angewandte Kognitionswissenschaft,
Universität Duisburg-Essen, Germany

Abstract

Robust time-to-contact calculation belongs to the most desirable techniques in the field of autonomous robot navigation. Using only image measurements it provides a method to determine when contact with a visible object will be made. However the computation of the time-to- contact values is very sensitive to noisy measurements of feature positions in a image. Instead of developing a new feature extraction and tracking algorithm this paper presents an approach which deals with the inaccurate measurements. It is based on the here derived equations which describe the process how a feature diverges from the focus of expansion. The results presented testify the stability and the robustness of this approach.

Keywords: TTC, structure from motion, robot navigation

1 Introduction

Robot obstacle and hazard detection is an important task within the field of robot navigation. It is fundamental to applications where successful and collision free robot navigation is required. 3D reconstruction of the surrounding environment is one possible solution to this problem.

In the case where the robot sensor system consists of video cameras only, visual information has to be used to obtain a three-dimensional structure or model of the world. This is a difficult task because the third dimension, the depth of the scene, has to be reconstructed from the two-dimensional images.

Numerous means of constructing such 3D models exist. Hartley et al. [2003] present a wide variety of stereo based algorithms for 3D reconstruction of the environment. Depending on how much prior information is available about the camera calibration and the relative positions of the cameras different algorithms can be applied to achieve different degrees of reconstruction. However these algorithms perform badly in the structure from motion approach if the only camera of the robot is headed in the direction of the robot motion and the robot is performing a forward movement. In this case the back projected rays are almost parallel for much of the field of view. This usually results in a poor reconstruction.

Since the described scenario is very common in the field of autonomous robot navigation different approaches exist to resolve this problem. A very promising technique is the estimation of the remaining time to contact with surrounding objects [Lee 1976], [Longuet-Higgins and Prazdny 1980], [van der Horst 1991], [Cutting et al. 1995], [van der Horst and Hogema 2003], [Hecht and Savelsbergh 2004]. Using only image measurements, and without knowing robot velocity or distance from the object, it is possible to determine when contact with a visible object will be made.

To achieve reliable time-to-contact values accurate measurement of feature positions in two consecutive images is of high importance. It has been shown that the quality of the time-to-contact values depends strongly on how precise the feature position in the image can be measured. This also corresponds to the conclusions made by Souhila et al.[Souhila and Karim 2007]. They have used an optical flow method to compute the point divergence from the focus of expansion. Bad light conditions caused large errors in the measurement and this resulted in unreliable time-to-contact values.

To resolve this problem much of the past research had been focused on developing algorithms for extracting outstanding features and for tracking them robustly ([Harris and Stephens 1988], [Shi and Tomasi 1994], [Lowe 2004], [Mikolajczyk and Schmid 2002]). However depending on the hardware used, image resolution and environmental conditions the required high accuracy of the measurement of the feature position can usually not be warranted.

This paper suggests a different approach. At first model equations are derived which describe how features diverge from the focus of expansion. Then based on the noisy measurements of feature positions the parameters of these equations are adapted in such a way that the true feature positions are best estimated. It will be shown that time-to-contact values which have been computed based on these estimated feature positions are much more stable and allow more reliable statements about the time to contact.

2 Calculating Time-to-Contact

This section discusses shortly the theory behind the calculation of the time-to-contact values and shows that the performance of this simple algorithm in an indoor environment is not satisfactory.

2.1 Theory behind the time-to-contact calculation

Time-to-contact calculation is widely used in the field of robotic vision. As stated before it allows to compute when contact with a visible object will be made by the robot. Knowledge of the robot velocity or its initial distance from the object is not required. However the approach works properly only in the case of a static environment. It also implies that the robot is moving with a constant velocity.

Good explanations of the theory behind time-to-contact can be found in [Trucco and Verri 1998], [Lee 1976] or [Camus 1995]. The following explanation overlaps largely with those made by Camus [1995].

Figure 1 describes the optical geometry for time-to-contact. A point of interest P at coordinates (X, Y, Z) is projected through the focus of projection centered at the origin of the coordinate system $(0, 0, 0)$. In physical space P is fixed and does not move. The origin or focus of projection, however move forward with a velocity $\frac{dZ}{dt}$. If the direction the camera is facing equals the direction of motion, then this direction is called the focus of expansion (FOE). In the case of a mobile robot it is quite reasonable to assume that

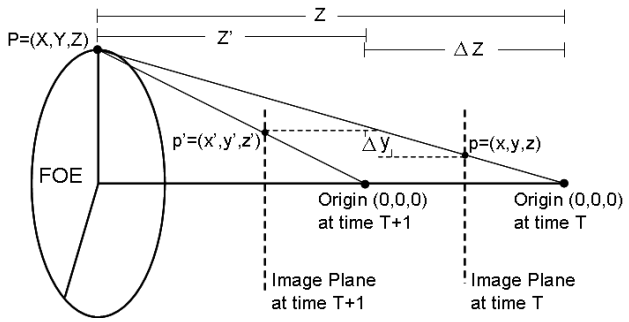

Figure 1: *Optical geometry for time-to-contact.*

the camera points the same direction as the direction of translation. The image plane is fixed at a distance z in front of the origin; for convenience we set $z = 1$. The actual value of z depends on factors such as the focal length of the camera. The world point P projects onto the point p in the image plane. When the robot is moving the image plane moves closer to P and the position of p in the image plane changes. Using equilateral triangles:

$$\frac{y}{z} = \frac{y}{1} = \frac{Y}{Z}$$

Differentiating with respect to time (where $\dot{a}$ represents the time derivative $\frac{da}{dt}$ for a given variable a):

$$\dot{y} = \frac{\dot{Y}}{Z} - Y\left(\frac{\dot{Z}}{Z^2}\right)$$

Since P is fixed, set $\dot{Y} = 0$. Now substituting (yZ) for Y:

$$\dot{y} = -y\left(\frac{\dot{Z}}{Z}\right)$$

Finally divide by y and take the reciprocals of both sides:

$$\frac{y}{\dot{y}} = -\frac{Z}{\dot{Z}} = \tau \qquad (1)$$

The quantity τ is known as the time-to-contact. Note that the computed τ does not give any information about distance or velocity per se, but only about their ratio. The above equation gives a method for calculating time-to-contact: for a camera heading in the same direction as the FOE, pick a point in the image, and divide its distance from the FOE by its divergence from the FOE.

Thus the algorithm for computing time-to-contact values consists of the following steps:

1. Compute the FOE from a sequence of consecutive images.
2. Find corresponding features in two consecutive images. For each of these correspondences time-to-contact values will be estimated.
3. Compute the lengths of the disparity vectors or optic flow vectors formed by two corresponding features. The lengths of the disparity vectors provide an estimation for the divergence of a given point from the FOE.
4. For every relevant feature compute its distance to the FOE and with this the time-to-contact value.

FOE calculation is based on the fact that it builds the center of the radial flow pattern which arises when the robot is moving forward. Usually FOE is computed as the intersection point of disparity vectors.

2.2 Details of the experimental setup

For experimental purpose the only camera (standard off the shelf web camera) of the robot was oriented in the direction of the robot movement. The robot was then programmed to perform a pure translational motion in a forward direction with a constant velocity for a given time interval.

For features to track SIFT Features (Scale Invariant Feature Transform) [Lowe 2004] have been chosen. Empirically it has been shown in [K. Mikolajczyk 2003] that they outperform most point detectors and are more resilient to image deformations. They are also robust to changes in illumination and noise.

For FOE calculation an initial sequence of seven consecutive images has been used. Figure 2 depicts graphically the idea behind the FOE calculation.

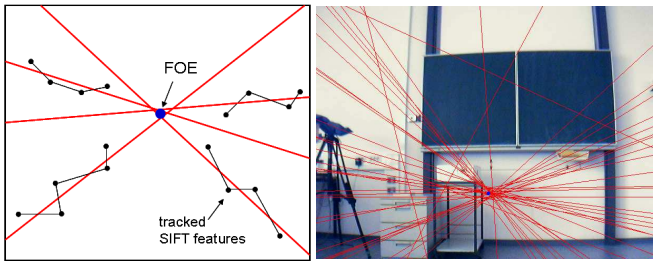

Figure 2: *FOE calculation: The left graphic shows the principal idea behind the FOE calculation. The right image shows the results of its implementation.*

After having tracked different SIFT features over a given sequence of consecutive images least mean squares (LMS) method has been used to find the optimal lines through these points. Then several intersection points of two randomly chosen lines have been computed and the average of these intersection points has been set as the FOE.

2.3 Results of the simple time-to-contact calculation

Figures 3 and 4 show exemplary the experimentation results of the described algorithm.

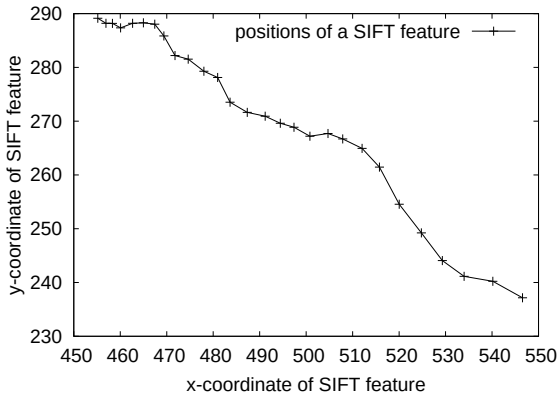

Figure 3: *Experimentation results: The graphic shows the positions of one selected feature which could have been tracked over 28 consecutive images.*

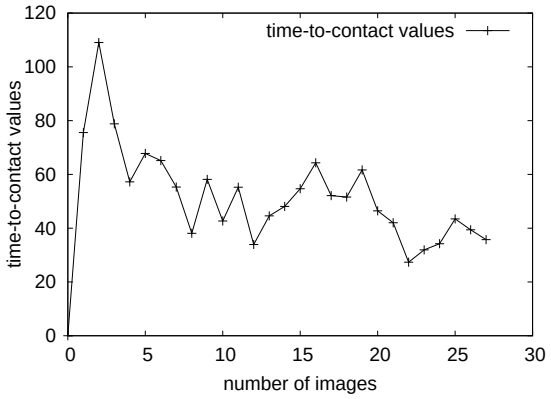

Figure 4: *Experimentation results: The graphic shows the time-to-contact values which correspond to the feature in figure 3.*

Figure 3 shows the positions of one randomly selected SIFT feature measured in every image of a sequence. The resulting time-to-contact values are shown in figure 4. The computed values provide no reliable information about the time to contact. Since the robot is approaching an object the computed time-to-contact values should be monotonic decreasing with respect to time. As one can see in figure 4 it is not the case.

Experimentally it was shown that the following reason has a major impact on the results: Due to small image resolution and inexact feature localisation the euclidean distances (disparities) between two consecutive feature positions do not result in a strictly increasing curve when plotted over time.

3 Method for robust time-to-contact calculation

The approach presented here aims the computation of time-to-contact values which allow more concrete statements. The main goal was to develop an algorithm which has the same noisy data as an input but produces more reliable time-to-contact values compared to those presented in the previous section.

The major idea behind the approach is that the tracked features do not diverge randomly from the FOE, but do follow a certain pattern.

3.1 Constructing a model

The feature position in the next image depends strongly on the 3D coordinates of the real world point and on the distance the robot drives before taking the next image. To demonstrate this relation, consider a 3D point P with the coordinates $P_{t1} = (X, Y, Z)$ relative to the robot position at timestamp $t1$. When the robot covers a distance d, only the Z-coordinate of the point P changes. Thus at timestamp $t2$ the point P has the coordinates $P_{t2} = (X, Y, Z - d)$ and at timestamp $t3$ the coordinates $P_{t3} = (X, Y, Z - d - d)$. Assuming a perspective camera model the 3D point P projects onto the following image points p at different timestamps.

$$
\begin{aligned}
p_{t1} &= (x_1, y_1) = \left(\tfrac{X}{Z}, \tfrac{Y}{Z}\right) \\
p_{t2} &= (x_2, y_2) = \left(\tfrac{X}{(Z-d)}, \tfrac{Y}{(Z-d)}\right) \\
p_{t3} &= (x_3, y_3) = \left(\tfrac{X}{(Z-d-d)}, \tfrac{Y}{(Z-d-d)}\right)
\end{aligned}
$$

Considering only the x-coordinates of these points one gets the following three equations:

$$X = x_1 \cdot Z \tag{2}$$

$$X = x_2 \cdot Z - x_2 \cdot d \tag{3}$$

$$X = x_3 \cdot Z - x_3 \cdot d - x_3 \cdot d \tag{4}$$

Equating (2) and (3):

$$Z = \frac{x_2 \cdot d}{(x_2 - x_1)} \tag{5}$$

Substituting equations (2) and (5) into (4) results in:

$$x_1 \left(\frac{x_2 \cdot d}{(x_2 - x_1)}\right) = x_3 \left(\frac{x_2 \cdot d}{(x_2 - x_1)}\right) - x_3 \cdot d - x_3 \cdot d$$

Rearranging the values:

$$d \cdot \frac{x_1 \cdot x_2}{(x_2 - x_1)} = d \cdot \left(\frac{x_3 \cdot x_2}{(x_2 - x_1)} - x_3 - x_3\right)$$

After canceling the variable d and further rearranging the values one gets the following final equation:

$$x_3 = \frac{x_1 \cdot x_2}{(2 \cdot x_1 - 1 \cdot x_2)}$$

Using complete induction (a method of mathematical proof) it can be shown that the above equation holds for every $n \in \mathrm{N}_{>0}$:

$$x\,(n) = \frac{x_1 \cdot x_2}{((n-1) \cdot x_1 - (n-2) \cdot x_2)} \tag{6}$$

The same considerations also hold for the y-coordinates of these image points:

$$y\,(n) = \frac{y_1 \cdot y_2}{((n-1) \cdot y_1 - (n-2) \cdot y_2)} \tag{7}$$

As one can see only values which can be measured directly from the image occur in these equations. Thus knowing the feature position in the first two images one can predict the feature position in the $(n-1)$-th and in the n-th image by applying the equations (6) and (7). By combining these values with FOE it is then possible to compute future time-to-contact values.

However experimentally it was shown that due to inexact feature localisation the computation of future time-to-contact values yields poor results. It is very sensitive to the first two measured positions of a feature. As stated above in most applications the precise measurement of feature positions can usually not be warranted.

3.2 Defining the optimization problem

Another way of interpreting the equations (6) and (7) is thinking of them as regression equations. Both equations describe the process of how a feature diverges from the FOE with every image taken from the camera. In other words they describe the relationships between the dependent variables x_n and y_n and the independent variable n. By setting $a_x = x_1$, $b_x = x_2$ and $a_y = y_1$, $b_y = y_2$ one gets:

$$x(n) = \frac{a_x \cdot b_x}{((n-1) \cdot a_x - (n-2) \cdot b_x)} \tag{8}$$

$$y(n) = \frac{a_y \cdot b_y}{((n-1) \cdot a_y - (n-2) \cdot b_y)} \tag{9}$$

Thus the task to solve is to find best estimate for the regression parameters a_x, b_x, a_y and b_y using positions of a given feature in the previously taken images. The following error functions $F(a_x, b_x)$ and $F(a_y, b_y)$ have to be minimized:

$$F(a_x, b_x) = \sum_{n=1}^{N} F_n(a_x, b_x), \quad F(a_y, b_y) = \sum_{n=1}^{N} F_n(a_y, b_y)$$

where $F_n(a_x, b_x)$ and $F_n(a_y, b_y)$ are defined as follows:

$$F_n(a_x, b_x) = \left(x_n - \frac{a_x \cdot b_x}{((n-1)\, a_x - (n-2)\, b_x)} \right)^2 \quad (10)$$

$$F_n(a_y, b_y) = \left(y_n - \frac{a_y \cdot b_y}{((n-1)\, a_y - (n-2)\, b_y)} \right)^2 \quad (11)$$

Here x_n and y_n are the measured coordinates of a given feature in the n-th image and N is the number of images taken until the actual timestamp.

By minimizing the error functions with every image taken from the camera one gets better and better estimation of the regression parameters and with it a better estimation of the true position of a given feature. Results show that time-to-contact values computed based on these estimated feature positions are more stable and allow more reliable statements about the robot relative distance to the objects.

3.3 Solving the optimization problem

The use of standard optimization methods (like gradient descent) to find the minimum of the functions defined above was shown to be very inefficient. With every image taken the error functions change and thus have to be optimized again.

To overcome this problem the stochastic gradient descent [Spall 2003] has been used. Here the true gradient is approximated by the gradient of the error function only evaluated on the recently observed position of a given feature. The parameters are then adjusted by an amount proportional to this approximate gradient. The update equations for the regression parameters have the following form:

$$a_x^{n+1} = a_x^n - \delta \left. \frac{\partial F_n(a_x, b_x)}{\partial a_x} \right|_{(a_x^n, b_x^n)}$$

$$b_x^{n+1} = b_x^n - \delta \left. \frac{\partial F_n(a_x, b_x)}{\partial b_x} \right|_{(a_x^n, b_x^n)}$$

Here the a_x^n and b_x^n are the latest estimations for the regression parameters a_x and b_x and $F_n(a_x, b_x)$ is defined as in equation 10. The variable δ is also updated with every image taken and has the value $\delta = \frac{0,1}{n^2}$. As initial values for the parameters a_x and b_x the first two x-coordinates of a given feature have been chosen: $a_x = x_1$ and $b_x = x_2$. This choice assures that the searched minimum lies in the near neighborhood.

The error function for the x-coordinates was then minimized under the following condition:

$$|a_x| < |b_x|$$

The condition implies that a feature diverges from the FOE. As the origin of the image coordinate system the calculated FOE was chosen.

The error function for the y-coordinates was minimized equivalently.

4 Results of the presented approach

Using the estimated regression parameters the positions of a feature in the n-th and the $(n-1)$-th image were calculated using the equations 8 and 9. Together with the computed FOE the time-to-contact values for every tracked feature were estimated using the equation 1. The achieved results are shown in figure 5 and 6.

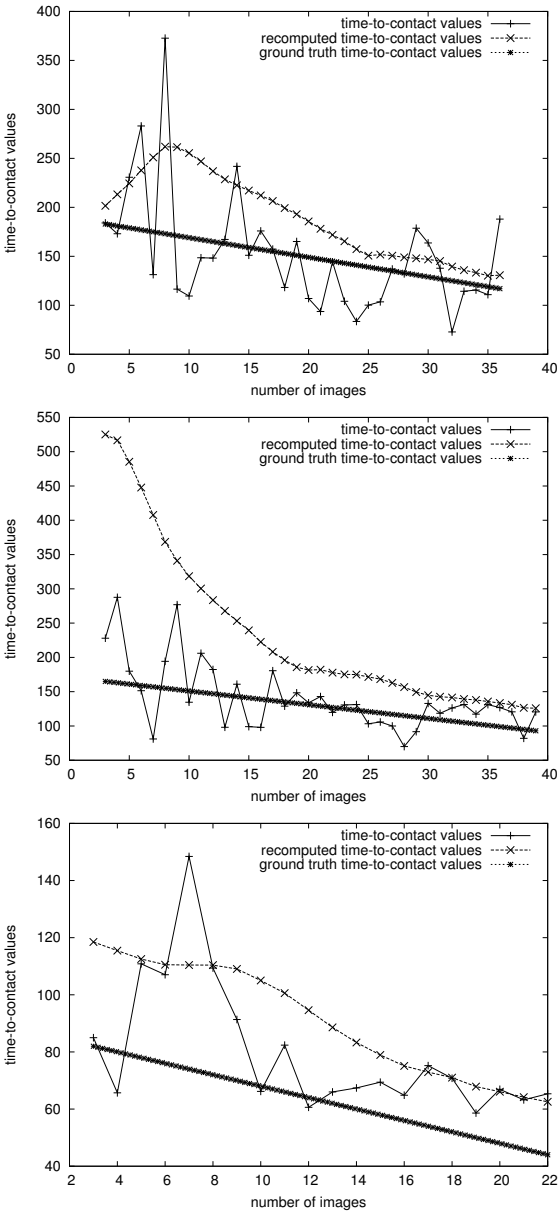

Figure 5: *Each graphic shows calculated time-to-contact values for one respective feature.*

In every graphic the time-to-contact values are plotted over the number of images the respective feature had been tracked. The continuous line shows the time-to-contact values which have been computed using the usual method described in Section 2. The dashed line shows the time-to-contact values which have been computed using the proposed approach. To gain some insight about the accuracy of the results also the ground truth time-to-contact values have been plotted on every graphic. Since the robot is moving forward

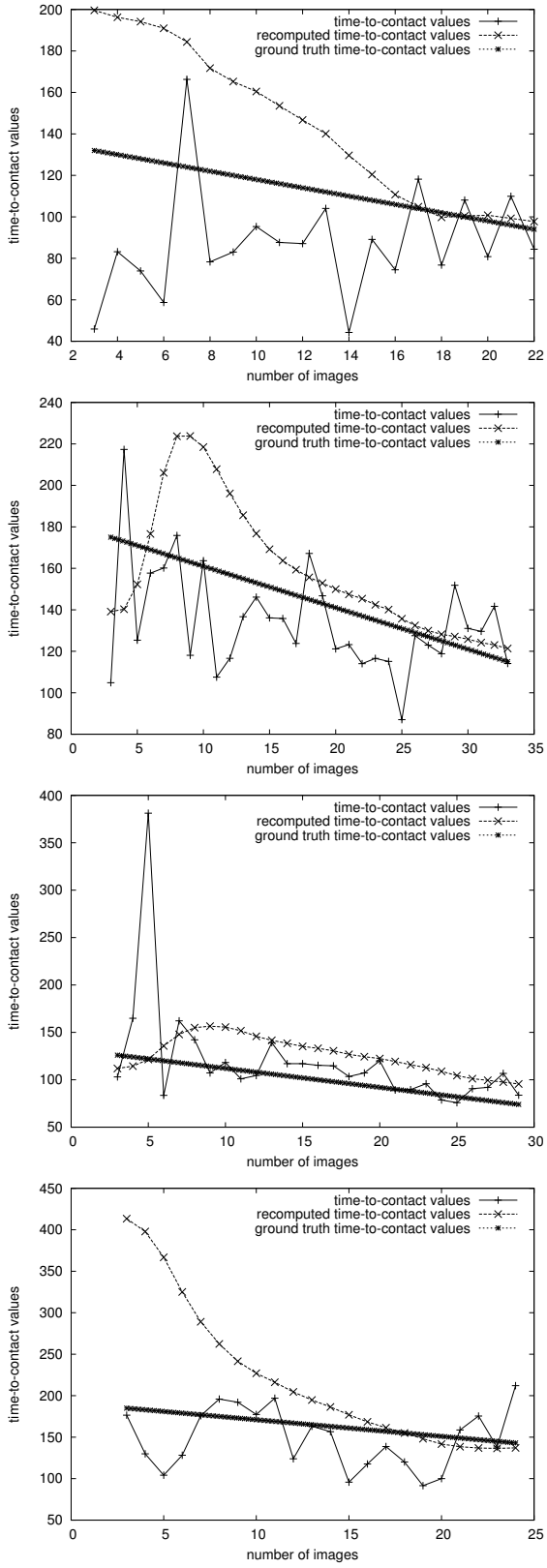

Figure 6: *Each graphic shows calculated time-to-contact values for one respective feature.*

with the constant velocity the ground truth values are represented through the straight lines.

As one can see the simple time-to-contact computation is not sufficient at all. The values jump from hight to low and at no point one can predict the further development of this curve. As the consequence no reliable statements about the time to contact can be made. In contrast the dashed line is more smooth and avoids sudden jumps. Moreover after a period of about ten images it starts to approximate the real time-to-contact values. During this initial period the regression parameters are adapted due to noisy measurement of feature positions.

Figure 7 shows a snapshot of a time-to-contact map calculated by the robot while moving forward. The upper figure shows the corresponding image taken from the camera.

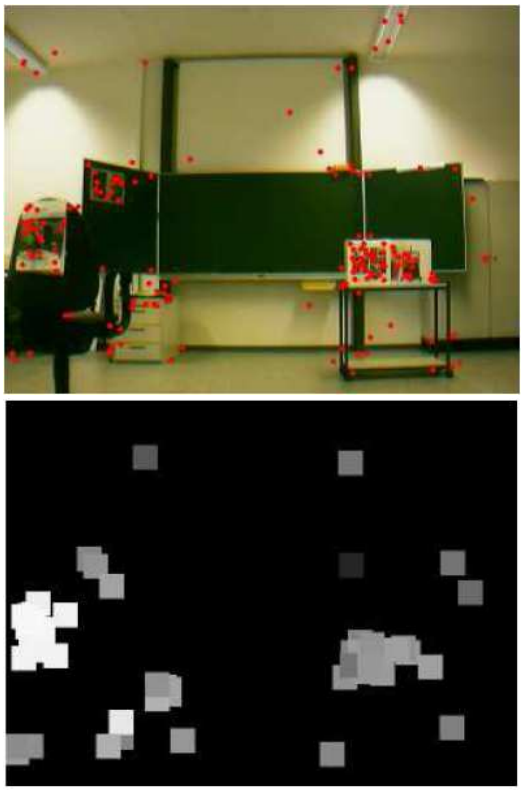

Figure 7: *Time-to-contact map: The upper image was taken from the camera and the lower image shows the corresponding time-to-contact map computed by the robot.*

The red points in the upper image depict the measured positions of tracked SIFT features. For every successfully tracked feature the corresponding time-to-contact values have been estimated. The results are shown in the lower image. Here the black color means that either no information is available or the object is relatively far away. The brighter the color the nearer is the respective object. As one easily realises the chair to the left is nearer than the box on the right side of the image. Thus on the time-to-contact map the chair is marked via the white spots and the box via the gray spots. Due to the smoothness of the curves which in figure 5 and 6 represent the recomputed time-to-contact values the computed time-to-contact map does not change abruptly from one image to the next. This is important if the results of the reconstruction should be used for planning tasks.

5 Conclusions

A new robust approach for calculating time-to-contact values has been presented. It is based on the here derived equations which describe how a feature diverges from the FOE. These equations are based only on the general rules about the image formation process and do not require any prior knowledge about the camera intrinsic parameters.

The thereon formulated error functions are minimized using the stochastic gradient descent optimisation algorithm. Due to this optimisation method it was shown that this approach is suitable for real time applications also.

It was successfully tested in an indoor environment with a robot which was equipped with a simple web cam. The results presented here testify the stability and the robustness of this approach.

6 Future Work

Although the presented method produce reliable time-to-contact values, it would be interesting to see how robust this approach is against low quality features or against false feature correspondences. Very important is also the extension of this approach to rotational movement of the robot.

References

CAMUS, T. 1995. Calculating time-to-contact using real-time quantized optical flow. *Max-Planck-Institut fuer biologische Kybernetik, Arbeitsgruppe Buelthoff, Technical Report*, 14.

CUTTING, J., VISHTON, P., AND BRAREN, P. 1995. How we avoid collisions with stationary and moving obstacles. *Psychological Review 4*, 102, 627–651.

FORSYTH, D., AND PONCE, J. 2003. *Computer Vision, A Modern Approach*. Pearson Education International.

HARRIS, C., AND STEPHENS, M. 1988. A combined corner and edge detector. *In 4th Alvey Vision Conference*, 147–151.

HARTLEY, R., AND ZISSERMAN, A. 2003. *Multiple View Geometry in Computer Vision*. Cambridge University Press.

HECHT, H., AND SAVELSBERGH, G. J. 2004. *Time-to-contact*. Elsevier.

K. MIKOLAJCZYK, C. S. 2003. A performance evaluation of local descriptors. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1615–1630.

LEE, D. 1976. A theory of visual control of braking based on information about time-to-collision. *Perception*, 5, 437–459.

LONGUET-HIGGINS, H., AND PRAZDNY, K. 1980. The interpretation of a moving retinal image. *In Proceedings of the Royal Society of London 208*, Series B, 385–397.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 2*, 60, 91–110.

MIKOLAJCZYK, K., AND SCHMID, C. 2002. An affine invariant interest point detector. *In European Conference on Computer vision (ECCV) 1*, 128–142.

SHI, J., AND TOMASI, C. 1994. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 593–600.

SOUHILA, K., AND KARIM, A. 2007. Optical flow based robot obstacle avoidance. *International Journal of Advanced Robotic Systems 4*, 1, 13–16.

SPALL, J. 2003. *Introduction to stochastic Search and Optimisation: Estimation, Simulation and Control*. John Wiley & Sons.

TRUCCO, E., AND VERRI, A. 1998. *Introductory Techniques for 3D Computer Vision*. Prentice-Hall Inc.

VAN DER HORST, R., AND HOGEMA, J. 2003. Time-to-collision and collision avoidance systems. *In Proceedings of the 6th ICTCT workshop Salzburg*.

VAN DER HORST, R. 1991. Time-to-collision as a cue for decision-making in braking. *Vision in Vehicles III, Elsevier Science Publishers B.V.*, 19–26.