Gradient-Based time to contact on paracatadioptric camera

Fatima Zahra Benamar, Sanaa El Fkihi, Cédric Demonceaux, E. Mouaddib, Driss Aboutajdine

GRADIENT-BASED TIME TO CONTACT ON PARACATADIOPTRIC CAMERA

F. Benamar[1,4] *S. El Fkihi*[2,1] *C. Demonceaux*[3] *E. Mouaddib*[4] *D. Aboutajdine*[1]

[1]LRIT, Unité associée au CNRST, URAC 29, FSR, Mohammed V-Agdal University, Rabat, Morocco
[2] RIITM, ENSIAS, Mohammed V-Souissi University, Rabat, Morocco
[3] Le2i, UMR CNRS 6306 University of Burgundy, Le Creusot, France
[4] MIS, University of Picardie Jules Verne, Amiens, France

fatima.zahra.benamar @ u-picardie.fr

ABSTRACT

The problem of time to contact or time to collision (TTC) estimation is largely discussed in perspective images. However, a few works have dealt with images of catadioptric sensors despite of their utility in robotics applications. The objective of this paper is to develop a novel model for estimating TTC with catadioptric images relative to a planar surface, and to demonstrate that TTC can be estimated only with derivative brightness and image coordinates. This model, called "gradient based time to contact", does not need high processing such as explicit estimation of optical flow and feature detection/or tracking. The proposed method allows to estimate TTC and gives additional information about the orientation of planar surface. It was tested on simulated and real datasets.

Index Terms— Time to contact, omnidirectional vision, obstacle avoidance, collision detection, mobile robotic.

1. INTRODUCTION

The time to contact or time to collision (TTC) is the time available to a robot before reaching an object. When a robot has a translation motion along X-axis, TTC equation is:

$$TTC = -\frac{X}{\frac{dX}{dt}} \tag{1}$$

where X is the distance between the sensor and the obstacle, and $\frac{dX}{dt}$ is the sensor velocity. An advantage of the TTC formulation is that it can be obtained from the image plane, without measuring real quantities (distance, velocity, ...).

Many approaches have been adopted for computing TTC in perspective vision. In optical flow based methods, we cite [9] which estimated TTC using particular sensor based on polar and log-polar representation. Besides, to avoid optical flow explicit computation problems, methods based closed contours have been proposed. Thus, in [1], TTC is estimated by tracking active contours, it requires segmentation, which is a quite difficult task. Another TTC category approaches is "gradient based TTC" [4]. They use spatial and temporal image brightness; and avoids "high level" processing (segmentation, explicit optical flow estimation,...).

In this work, we propose to estimate TTC using a catadioptric camera. We may classify the existing works in two axes: (1) TTC estimated on the catadioptric image plane. In our previous work [2], we proposed a TTC formula at each point in the image plane using parabolic projection and adapted optical flow [8]. (2) TTC estimated on the unit sphere. For instance, [3] describes a closed form bound of TTC from local motion field in the case of rigid motion and surface orientation with an arbitrary wide FOV. Recently, [5] estimates TTC using the maximum flow field divergence on the view sphere. In our paper, we propose to compute the TTC directly in the image plane using "gradient based TTC". In section 2, we explain the discussed problem. Then, section 3 presents a new formula of TTC for omnidirectional images while the experimental results with synthetic and real sequences are given in section 4. Finally, conclusion is drawn in section 5.

2. MATHEMATICAL FORMULATION

The traditional gradient based TTC method of [4] is put forward in case of translation motion along the optical axis towards an arbitrary orientation surface and extended on the geometric modeling in [6]. A planar surface (using perspective projection) in image coordinates (x, y) is:

$$Z(x,y) = \frac{Z_0}{1 - a\frac{x}{f} - b\frac{y}{f}} \tag{2}$$

with f: focal length, Z_0: distance to the surface along Z-axis (optical axis), and (a, b): slopes in (X, Y) directions.

The brightness constancy constraint equation (BCCE) can be written:

$$UE_u + VE_v + E_t = 0 \tag{3}$$

with (E_u, E_v, E_t) are the brightness derivatives $w.r.t\ (x, y, t)$. $(U, V) = (\alpha_u \frac{dx}{dt}, \alpha_v \frac{dy}{dt})$ is the optical flow in the image and:

This work was supported by STIC program and CNRST.

$$\frac{dx}{dt} = -x\frac{\dot{Z}}{Z} \quad ; \quad \frac{dy}{dt} = -y\frac{\dot{Z}}{Z} \tag{4}$$

while (α_u, α_v) is a scale conversion mm to pixel in the (x, y) direction. Using eq.(2), eq.(3), eq.(4). The TTC is solution of:

$$G(C + Px + Qy) + E_t = 0 \tag{5}$$

where, $C = -\dot{Z}/Z_0$ is the inverse of TTC, $P = -(\frac{a}{f})C$, $Q = -(\frac{b}{f})C$, G is the radial gradient $(\alpha_u x E_u + \alpha_v y E_v)$.

Nonetheless, using paracatadioptric camera, the mirror introduces additional geometric transformations. Indeed, scene points are projected onto the image plane according to a non-linear model. So, eq.(2), eq.(4) and eq.(5) are not valid. We have to introduce parabolic projection into the surface equation to express the depth function in the image coordinates.

Let us consider the case of paracatadioptric sensor; the equation of the parabolic mirror is given by:

$$z_p = \frac{x_p^2 + y_p^2 - h^2}{2h}. \tag{6}$$

with h is a mirror parameter. Let $P(X, Y, Z)$ be a 3D point. P is projected onto the mirror on the point $m_p(x_p, y_p, z_p)$. Then, m_p is projected on the camera plane on $m(x, y)$ (see figure 1). In pixel coordinates, the image point $m(u, v)$ is given by eq.(8) ((u_0, v_0): a principal point coordinates).

$$\begin{cases} x_p = \frac{h\ X}{\sqrt{X^2+Y^2+Z^2}-Z} = x \\ y_p = \frac{h\ Y}{\sqrt{X^2+Y^2+Z^2}-Z} = y \\ z_p = \frac{h\ Z}{\sqrt{X^2+Y^2+Z^2}-Z} \end{cases} \tag{7}$$

$$u = \alpha_u x + u_0 \quad ; \quad v = \alpha_v y + v_0 \tag{8}$$

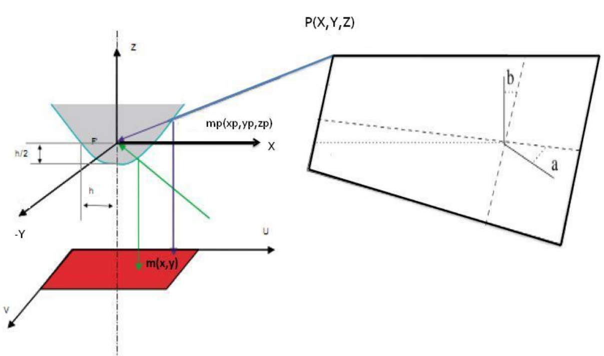

Fig. 1. Configuration of paracatadioptric projection

A given planar surface (see figure 1) is defined by:

$$X(Y, Z) = X_0 + aY + bZ \tag{9}$$

where a and b give the planar surface slopes with respect to X-axis, while X_0 is the distance robot-surface along X-axis.

To define the depth of a surface in a paracatadioptric image, we use eq.(6), eq.(7), and eq.(9). So, a planar surface can be expressed as a function of the image coordinates (x, y):

$$X(x, y) = \frac{X_0}{1 - (\frac{ay}{x} + \frac{b(x^2+y^2-h^2)}{2hx}))} \tag{10}$$

In the next section, we will define a new equation which links TTC to m, to take into account the mirror distortion.

3. A NEW FRAMEWORK TO ESTIMATE TTC

We consider paracatadioptric sensor moving along the X-axis towards an inclined surface. Let $P(X, Y, Z)$ be a fixed obstacle in the environment space $\mathbb{R}^3$. In the case of the perspective camera, TTC is defined by the time it would take for the object to cross the image plane. But, in the case of paracatadioptric camera, we define the collision condition when the obstacle crosses the parabolic mirror focus. The contact point on the plane surface is $(X_0, 0, 0)^T$ and its TTC equation is:

$$TTC = -\frac{X_0}{\dot{X}} \tag{11}$$

We set $C_X = -\frac{\dot{X}}{X}$. Eq.(11) can be written as:

$$TTC = \frac{1}{C} = -\frac{X_0}{X}.\frac{X}{\dot{X}} = \frac{X_0}{X}.\frac{1}{C_X} \tag{12}$$

Hence, from eq.(10) and eq.(12), we obtain:

$$C_X = C.(1 - (a\frac{y}{x} + b\frac{x^2 + y^2 - h^2}{2hx})) \tag{13}$$

The key idea of the new method is the exploitation of the BCCE. We compute the image coordinate derivatives x and y from eq.(7), we consider $\alpha_u = \alpha_v = \alpha$, and we set: $A = X^2 + Y^2 + Z^2$. Thus, we obtain:

$$\begin{cases} \dot{x} = xC_X(\frac{x}{h}(XA^{-1/2}) - 1) \\ \dot{y} = C_X(\frac{xy}{h}(XA^{-1/2})) \end{cases} \tag{14}$$

From eq.(6) and eq.(7); and $z_p + h > 0$, we have:

$$XA^{-1/2} = \frac{x}{(z_p + h)} = \frac{2hx}{x^2 + y^2 + h^2} \tag{15}$$

Therefore, using (13), $\dot{x}$ and $\dot{y}$ are written as:

$$\begin{cases} \dot{x} = xC.(1 - (a\frac{y}{x} + b\frac{x^2+y^2-h^2}{2hx}))(\frac{2x^2}{x^2+y^2+h^2} - 1) \\ \dot{y} = yC.(1 - (a\frac{y}{x} + b\frac{x^2+y^2-h^2}{2hx}))(\frac{2x^2}{x^2+y^2+h^2}) \end{cases} \tag{16}$$

From eq.(3), eq.(8) and eq.(16), we obtain the following equation in pixel coordinates:

$$S = C[\frac{2(u-u_0)^2}{(u-u_0)^2 + (v-v_0)^2 + (\alpha h)^2}G^* - (u-u_0)E_u] +$$

$$P[\frac{2(u-u_0)(v-v_0)}{(u-u_0)^2+(v-v_0)^2+(\alpha h)^2}G^* - (v-v_0)E_u]+$$

$$+Q[\frac{(u-u_0)}{\alpha.h}\frac{(u-u_0)^2+(v-v_0)^2-(\alpha h)^2}{(u-u_0)^2+(v-v_0)^2+(\alpha h)^2}G^* -$$

$$\frac{(u-u_0)^2+(v-v_0)^2-(\alpha h)^2}{2.\alpha.h}E_u] + E_t = 0 \qquad (17)$$

with $P = -aC$, $Q = -bC$ and $G^* = ((u-u_0)E_u + (v-v_0)E_v)$. Eq.(17) has to estimate: C (inverse of TTC), P and Q. To solve this equation, we proceed as follows:

- We consider a region of interest Δ in the image. For all points in Δ, we write its corresponding eq.(17);
- We construct the system of the obtained equations composed of as many equations as points in the Δ;
- We solve this linear system by the least squares method.

We can rewrite eq.(17) as:

$$H\Theta = B \qquad (18)$$

where

$$\Theta = \begin{pmatrix} C \\ P \\ Q \end{pmatrix};\ H = [H_1 \quad H_2 \quad H_3];\ B = [-E_t] \qquad (19)$$

$$H_1 = \frac{2(u-u_0)^2}{(u-u_0)^2+(v-v_0)^2+(\alpha h)^2}G^* - (u-u_0)E_u \qquad (20)$$

$$H_2 = \frac{2(u-u_0)(v-v_0)}{(u-u0)^2+(v-v_0)^2+(\alpha h)^2}G^* - (v-v_0)E_u \qquad (21)$$

$$H_3 = \frac{(u-u_0)}{\alpha.h}\frac{(u-u_0)^2+(v-v_0)^2-(\alpha h)^2}{(u-u_0)^2+(v-v_0)^2+(\alpha h)^2}G^* -$$

$$\frac{(u-u_0)^2+(v-v_0)^2-(\alpha h)^2}{2.\alpha.h}E_u \qquad (22)$$

Remark, if $a = b = 0$, that is, if the planar surface is perpendicular to the X-Axis, the eq.(17) becomes:

$$CH_1 = -E_t \qquad (23)$$

Based on spatial gradients, temporal gradient, image coordinates and constants α, h and (u_0, v_0), our approach allows to estimate TTC and the orientations of a planar surface a and b.

4. EXPERIMENTAL RESULTS

Here, we will evaluate the new TTC proposal for perpendicular and inclined surfaces, in both of synthetic and real images.

4.1. Synthetic sequence

Our synthetic sequence is generated by Pov-ray software[1]. Virtual paracatadioptric sensor is moving with uniform unidirectional motion towards X-axis (see figure 2) and generates images of $200*200$ pixels. For all sequences, we have: α=40, h=2.5 and (u_0, v_0)=$(100, 100)$. We denote by (β, θ) the planar surface inclination angles around (Z-axis,Y-axis); where a=$\frac{\tan(\beta)}{\cos(\theta)}$ and b=$-\tan(\theta)$. Figure 3 shows TTCs (red line) of four sequences with inclination angles $(\beta, \theta) = (0°, 0°)$, $(-10°, -5°)$, $(-20°, -10°)$ and $(-30°, -20°)$. The ground truth is illustrated in dashed green line. In this figure, it is shown that our algorithm gives interesting results, TTC values dropped linearly as expected.

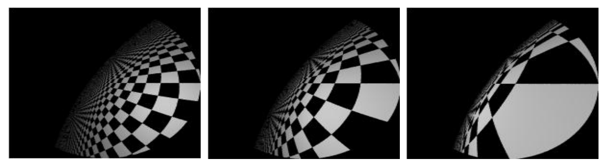

Fig. 2. Synthetic image of inclined planar surface with $\beta = -30°$ and $\theta = -20°$; (left to right) frame $n°1$, 100, and 170.

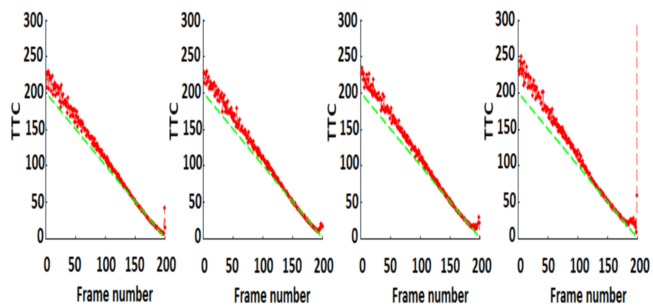

Fig. 3. Estimation of TTC on synthetic sequences for different inclinations of the planar surface.

The commonly used performance measure between an estimated vector (V_e) and a real one $(V_r$ is angular error (AE):

$$AE = \arccos(\frac{< V_e, V_r >}{||V_e|| ||V_r||}) \qquad (24)$$

Let us consider the vector $(-1, a, b)$, figure 4 shows the evaluation results of a and b for the considered synthetic sequences above. Because of the bias introduced by b, AE is larger at the beginning of the sequence. We can note some perturbations at the beginning and at the end of the sequence. This phenomena is well known in optical flow estimation and due to temporal aliasing. The latter can be overcome using a multiresolution framework [4] after alteration for paracatadioptric images[7].

[1] www.povray.com

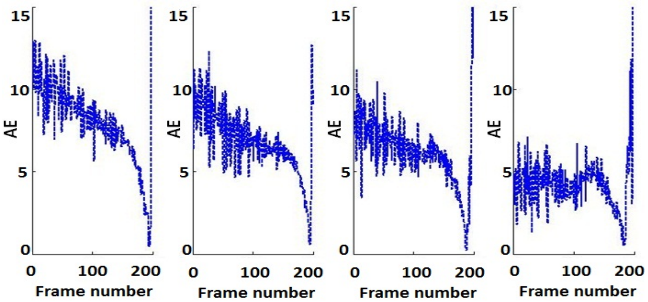

Fig. 4. Angular error (AE) in degree for synthetic sequence. From left to right: AE corresponds to sequence with $(\beta, \theta) = (0°, 0°)$, $(-10°, -5°)$, $(-20°, -10°)$ and $(-30°, -20°)$.

4.2. Real sequence

We consider a paracatadioptric sensor embedded on the Pioneer robot. It generates images of $480 * 600$ pixels for all sequences. The camera calibration was conducted with the Hyscac toolbox[2] and the intrinsic parameters are: (u_0, v_0)=$(237.13, 316.04)$, h=1, and α=156.39. The sensor moves along X-axis towards a perpendicular wall (rigid surface) (see figures 5.(a) and (b)). It has also acquired sequences with inclined wall (see figures 5.(c) and (d)). An online demo of this work is available at [3].

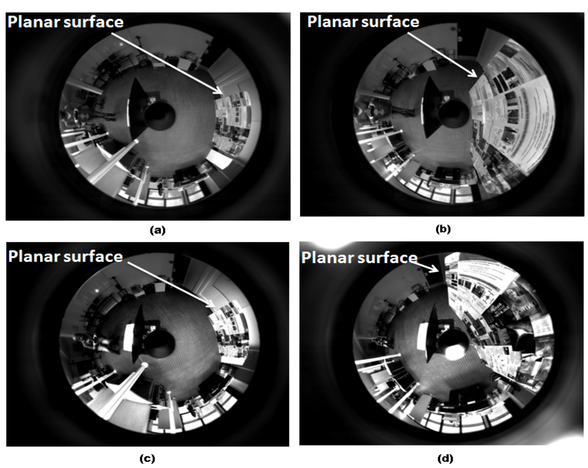

Fig. 5. Images of two real sequences; (a) and (c) are the first frames while (b) and (d) are the last ones. (a) and (b): Images with perpendicular planar surface. (c) and (d): Images with inclined planar surface.

Figures 6 and 7 show the TTC computed with our proposal algorithm. As the robot speed is constant during the experiment, the TTC should decrease linearly (dashed green line in figures 6 and 7). TTC curve decreases linearly from the first to the last frame as expected. The robot is able to stop before collision at a security distance of 21 frames.

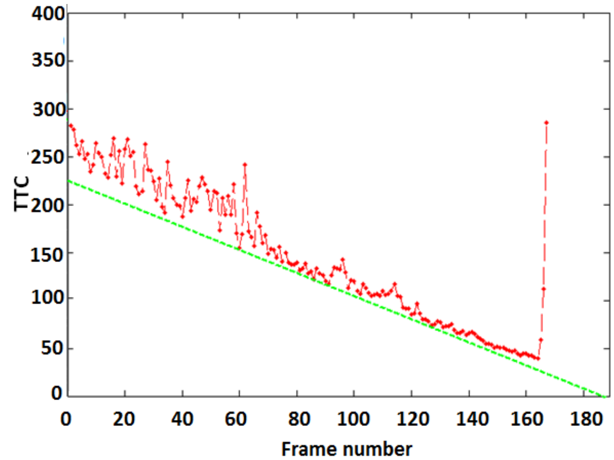

Fig. 6. Estimation of TTC (red line) on real sequence in the case of perpendicular planar surface.

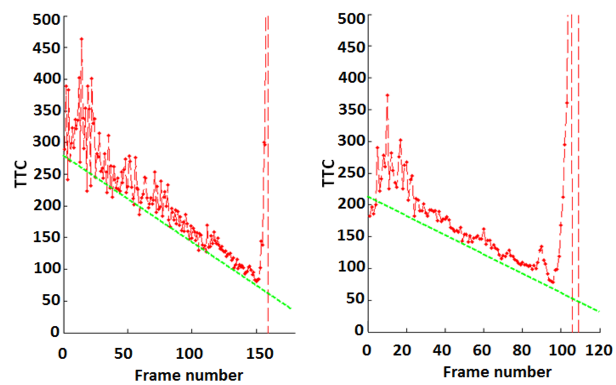

Fig. 7. Estimation of TTC (red line) on real sequence in case of two different inclined surfaces.

5. CONCLUSION AND FUTURE WORK

In this paper we proposed a novel framework to estimate TTC in omnidirectional images for the planar surface of an arbitrary orientation. The introduced approach gives additional information about the planar surface. Focus was on gradient based methods because they are simple and fast, and they avoid high level processing in computer vision, thus making them a good candidate for real time implementations. The illustrated results show that our approach has the potential to be effective in complex real-world scenarios. In fact, the work can be expanded in several directions: it could be interesting to generalize it to a general motion towards X and Y axes.

[2]http://www.hyscas.com

[3]http://home.mis.u-picardie.fr/∼fatima

6. REFERENCES

[1] G. Alenyà, A. Nègre, and J. L. Crowley. A comparison of three methods for measure of time to contact. In *Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, IROS'09, pages 4565–4570, Piscataway, NJ, USA, 2009. IEEE Press.

[2] F.Z. Benamar, C. Demonceaux, S. EL. Fkihi, E. Mouaddib, and D. Aboutajdine. Time to contact estimation on paracatadioptric cameras. In *Pattern Recognition (ICPR), 2012 21th International Conference on*, number 3602 - 3605, November 2012.

[3] C. Colombo. Time to collision from first-order spherical image motion. *Robotics and Autonomous Systems*, 31(1-2):5 – 15, 2000.

[4] B.K.P. Horn, Y. Fang, and I. Masaki. Hierarchical framework for direct gradient-based time-to-contact estimation. In *In IEEE Intelligent Vehicles Symposium*, pages 1394–1400, Shaanxi, China, June 2009.

[5] C. McCarthy and N. Barnes. A unified strategy for landing and docking using spherical flow divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:1024–1031, 2012.

[6] F. G. Meyer. Time-to-collision from first-order models of the motion field - robotics and automation. *IEEE Transactions on*, 10(6), 1994.

[7] Y. Mochizuki and A. Imiya. Multiresolution optical flow computation of spherical images. In *Computer Analysis of Images and Patterns*, pages 348–355, January 2011.

[8] A. Radgui, C. Demonceaux, E. Mouaddib, M. Rziza, and D. Aboutajdine. An adapted lucas-kanade's method for optical flow estimation in catadioptric images. In *In OMNIVIS'2008, the Eighth Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, Marseille, France, October 2008.

[9] M. Tistarelli and G. Sandini. On the advantages of polar and log-polar mapping for direct estimation of time-to-impact from optical flow. *IEEE Trans. on PAMI*, 15:401–410, 1992.