Visual Contact with Catadioptric Cameras: A Real-Time Implementation in ROS

Devesh Adlakha, Cédric Demonceaux
Université de Bourgogne Franche-Comté
12 rue de la Fonderie, 71200 Le Creusot

El Mustapha Mouaddib
Université de Picardie Jules Verne (UPJV)
33 rue Saint Leu, 80039 Amiens Cedex 1

Abstract

Time to contact (TTC) is the time remaining till collision with an obstacle. The TTC estimation is well studied and established for perspective images. Recent work [1] has extended the formulation to catadioptric vision. This paper presents a real-time implementation of their optical flow-based method for TTC estimation using a catadioptric sensor on a mobile robot. The method is introduced, followed by the implementation details and experimental results. A Pioneer 3-AT mobile robot was used in the experiments, and development carried out in the framework of ROS.

1. Introduction

The time to contact with an obstacle is defined as the ratio of the distance till the obstacle and the velocity of motion with respect to the obstacle.

$$\tau = -\frac{Z}{\dot{Z}} \tag{1}$$

where Z is the distance till the obstacle, and $\dot{Z}$ is the velocity of motion.

In a vision-based approach, the time to contact may be obtained directly from the stream of images. For a perspective camera, and for the case of translation motion along the optical axis of the camera, the TTC is formulated as in [2]:

$$\tau = \frac{x}{\dot{x}} = \frac{y}{\dot{y}} \tag{2}$$

where (x, y) is an image coordinate, *i.e.* with respect to the principal point in the image, and $(\dot{x}, \dot{y})$ is the corresponding optical flow field.

In pixel coordinates,

$$\tau = \frac{u - u_0}{\dot{u}} = \frac{v - v_0}{\dot{v}} \tag{3}$$

Note that τ is in terms of the number of frames remaining before collision with an obstacle is imminent.

1.1. TTC in Catadioptric Vision

The TTC estimation based on optical flow is valid for central catadioptric cameras, and relies on the following relation:

$$\frac{x}{y} = \frac{X}{Y} \tag{4}$$

Deriving this relation leads to the formulation for catadioptric vision. For translation along the X-axis (parallel to the u axis of the camera), the formulation is given as:

$$\tau^{-1} = \frac{\dot{y}}{y} - \frac{\dot{x}}{x} = \frac{\dot{v}}{v - v_0} - \frac{\dot{u}}{u - u_0} \tag{5}$$

Similarly, for translation along the Y-axis (parallel to the v axis of the camera):

$$\tau^{-1} = \frac{\dot{x}}{x} - \frac{\dot{y}}{y} = \frac{\dot{u}}{u - u_0} - \frac{\dot{v}}{v - v_0} \tag{6}$$

2. Implementation

The Pioneer 3-AT mobile robot with a mounted catadioptric sensor is shown in fig. 1. Development was carried out using the roscpp client library in the ROS Hydro Medusa distribution.

2.1. Optical Flow Estimation

The literature on optical flow estimation techniques is dense, and [1] used the classical Lucas-Kanade optical flow algorithm [3]. For this implementation, the Gunnar-Farnebäck algorithm [4] was used, as it was already implemented in OpenCV, which provided a convenient and existing interface with ROS. Qualitative analysis showed the results obtained to be similar to those in [1] for a set of similar synthetic data images.

Figure 1: Pioneer 3-AT mobile robot with mounted catadioptric sensor.

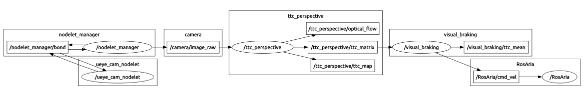

Figure 2: Graph of implementation design.

2.2. Design

The implementation is designed to be modular, with the computation of the TTC maps distinguished from the application. This is in part driven by the unspecified application of the resultant TTC maps. As shown in fig. 2, the ttc_perspective node subscribes to the image stream from the camera, and computes the TTC matrix along with the optical flow and TTC maps for visualization. The application node, visual_braking subscribes to the TTC matrix and utilizes it for robot control. Nodes for both perspective and catadioptric TTC map computations are included in the implemented package.

The optical flow algorithm parameters determine the TTC estimation and processing time. User configuration of these, as well the visualization parameters is supported in the following two ways:

- Configuration files: the desired parameters may be edited in a provided configuration file. Consequently, multiple such parameter configurations may be maintained for different experiments or applications, and the desired configuration to be used need only be specified before execution (in a launch file in ROS).
- Dynamic reconfiguring: the parameters may be be tuned dynamically through a graphical user interface in ROS. The parameters permitting dynamic reconfiguration, along with their default values and accepted range have been set manually. The parameters tuned through this method could subsequently be stored in a configuration file, and launched as mentioned above.

Figure 3: Obstacle in the environment used in the experiments.

3. Results

The obstacle in the environment used for the purpose of the experiments is shown in fig. 3. The arrangement of the obstacle was deliberate to introduce areas of high texture in order to achieve more accurate optical flow estimation.

The formulation in eq. (6) was used due to the translation motion along the $v-$ axis of the camera. Three frames from a conducted experiment corresponding to the image stream from the catadioptric camera and its resultant TTC map are shown in fig. 4. The colormap scale used in the visualization is shown adjacent to the images. The region of interest is the obstacle depicted in fig. 3. As the mobile robot approaches the obstacle, the TTC around the region gradually decreases, as indicated by the color shift from red to blue within the three frames shown. There appear a large number of outliers in the TTC estimation, shown in the dark red pixels, due to erroneous or minute optical flow estimates for these pixels. Note that the translation motion of the mobile robot towards the obstacle was manually controlled through a constant command velocity.

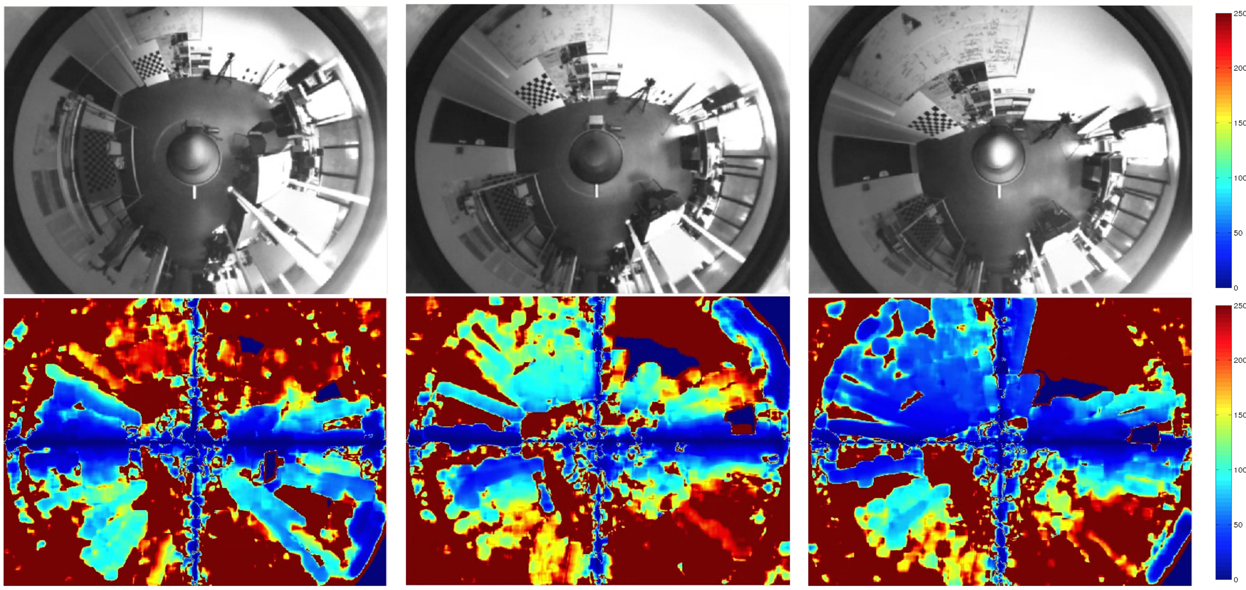

Figure 4: Results: TTC maps.

4. Conclusion

The optical flow-based TTC estimation method from [1] was implemented for real-time performance on a mobile robot. The implementation was carried out in the framework of ROS, contained in a package with nodes for both perspective and catadioptric methods, as well as two simple applications of visual braking and depth computation. Modularity was a design objective, and the computation of the TTC matrix is distinguished from the application in this regard. The user configuration of algorithm and visualization parameters, including dynamically, allow for swift experimentation.

5. Future Work

Although the method has been validated on a mobile robot system, its practical viability and use remains to be determined. The strong constraints of translation motion, and reliance on optical flow estimation render the method inferior to conventional laser range scanners for applications such as collision avoidance. In the short-term, the implementation in ROS is to be refined, and documented on the official wiki.

References

[1] F. Benamar, S. Elfkihi, C. Demonceaux, E. Mouaddib, and D. Aboutajdine, “Visual contact with catadioptric cameras,” *Robotics and Autonomous Systems*, vol. 64, pp. 100–119, 2015 (cit. on pp. 1, 3).

[2] T. Camus, “Calculating time-to-contact using real-time quantized optical flow,” *National Institute of Standards and Technology NISTIR*, vol. 5609, p. 51, 1995 (cit. on p. 1).

[3] B. D. Lucas, T. Kanade, *et al.*, “An iterative image registration technique with an application to stereo vision.,” in *IJCAI*, vol. 81, 1981, pp. 674–679 (cit. on p. 1).

[4] G. Farnebäck, “Two-frame motion estimation based on polynomial expansion,” in *Image Analysis*, Springer, 2003, pp. 363–370 (cit. on p. 1).