Moving object detection, tracking and following using an omnidirectional camera on a mobile robot

I. Marković, F. Chaumette and I. Petrović, in Proc. of IEEE International Conference on Robotics and Automation (ICRA'14), Hong Kong, 2014, pp. 5630-5635

Outline

1. Introduction
2. Camera calibration
3. Object detection
4. Object tracking on the sphere
5. Object following
6. Experiment results
7. Conclusion

Introduction

- Omnidirectional cameras are powerful in mobile robot’s vision system, due to it stored all information about the surrounding scene
- Panoramic videos provide more information than conventional video, it increase the robot’s abilities to react in the environment
- Combine omnidirectional cameras and mobile robot, for example, removing the possibility that the followed object will escape the camera’s field-of-view
- However, detecting, tracking, and following the moving object with a camera mounted on a mobile robot is challenging
- Because of the motion of the simultaneous ego-motion of the robot and the motion of the object

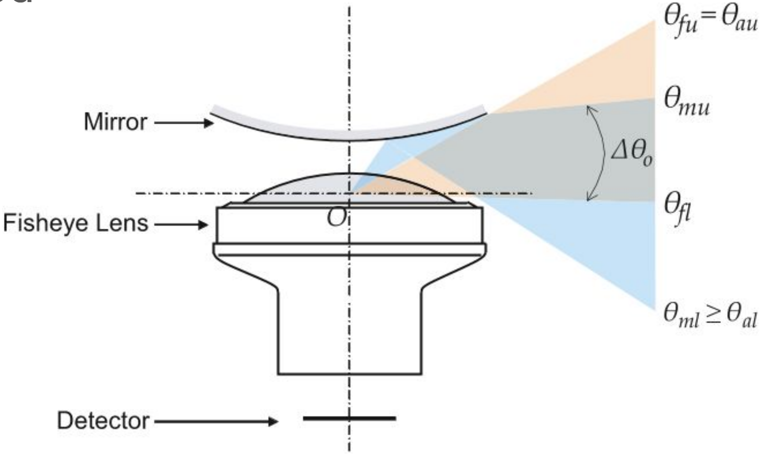
[1] http://ieeexplore.ieee.org/document/4544035/

Camera calibration

- An omnidirectional camera, which contains a fisheye lens (a wide-angle lens) with a catadioptric lens (a mirror) called catadioptric system [1]
- It mounted on a mobile robot to obtain the panoramic image
- Pros: decrease the stitching issues at the edges of the projection
- Cons: a unique calibration method is needed


(c)

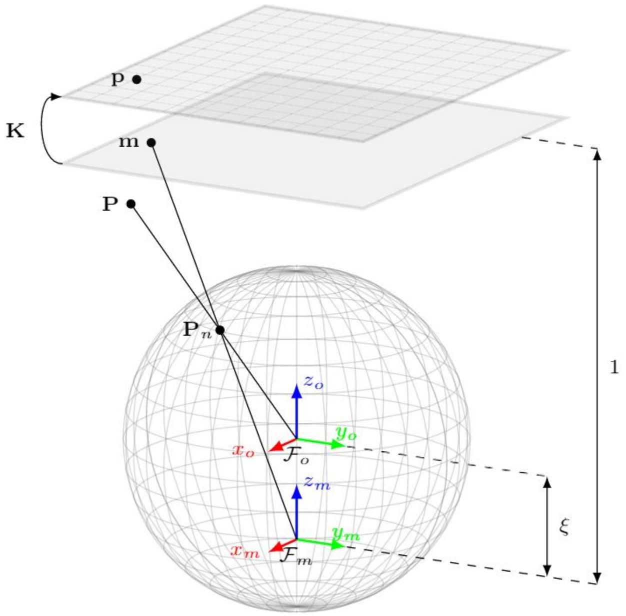

(d)



Camera calibration (cont.)

- Single view point omnidirectional camera calibration from planar grids
- K is a 3x3 matrix containing the camera intrinsic parameters
- K and ξ are obtained by the calibration procedure
- planar -> sphere

$$\mathbf{m} = \mathbf{K}^{-1}\mathbf{p}, \quad \mathbf{P}_n = \begin{bmatrix} \frac{\xi + \sqrt{1 + (1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} x \\ \frac{\xi + \sqrt{1 + (1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} y \\ \frac{\xi + \sqrt{1 + (1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} - \xi \end{bmatrix}. \quad (1)$$



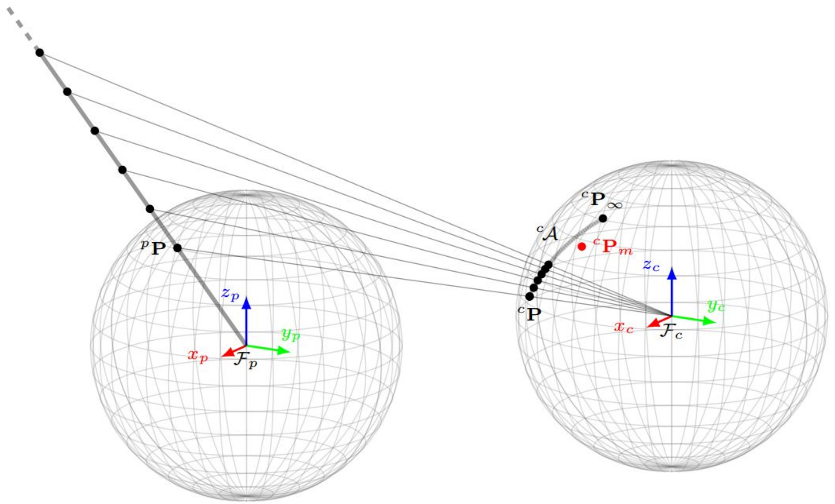
[2] https://en.wikipedia.org/wiki/Lucas%E2%80%93Kanade_method
[3] http://ieeexplore.ieee.org/document/6046621/

Object detection

- Detect moving object in the omnidirectional image while the robot itself moves
- Motion in the image is caused by moving object and ego-motion of robot
- To figure out this challenge, they calculate the sparse optical flow (Lucas-Kanade algorithm [2]) in the image, which caused by the ego-motion and moving object [3]
- Sparse techniques only process some pixels from the whole image

Object detection (cont.)

- Now, we need to find the optical flow caused by the moving object only
- $\mathcal{F}_p$: previous frame
- $\mathcal{F}_c$: current frame
- ${}^c\mathbf{R}_p$: rotation
- ${}^c\mathbf{t}_p$: translation
- ${}^p\mathbf{P}$: point in prvious frame
- ${}^c\mathbf{P}_m$: point in current frame



- First, hypothesize the flow was ego-motion, if the condition is not met, it should be caused by the moving object

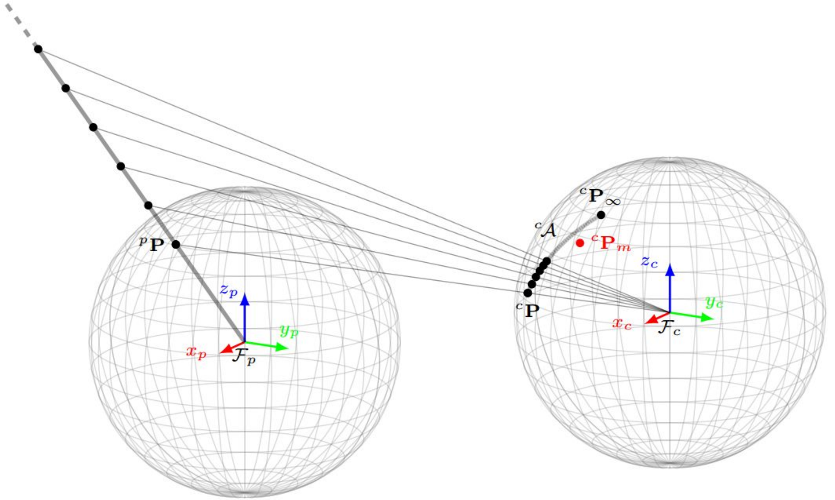
Object detection (cont.)

- $^{c}\mathbf{P}_{\infty} = {}^{c}\mathbf{R}_{p}{}^{p}\mathbf{P}$
- $^{p}\mathbf{P}$ represent a point on the previous sphere
- If an optical flow as caused by ego-motion if its matched point on the current sphere $^{c}\mathbf{P}_{m}$ lies somewhere along the arc of the great circle
- Great circle distance

$$d({}^{c}\mathbf{P}, {}^{c}\mathbf{P}_{\infty}) = \arccos\left({}^{c}\mathbf{P} \cdot {}^{c}\mathbf{P}_{\infty}\right), \qquad (2)$$

- Calculate the distance of $^{c}\mathbf{P}_{m}$ to arc $^{c}\mathcal{A}$

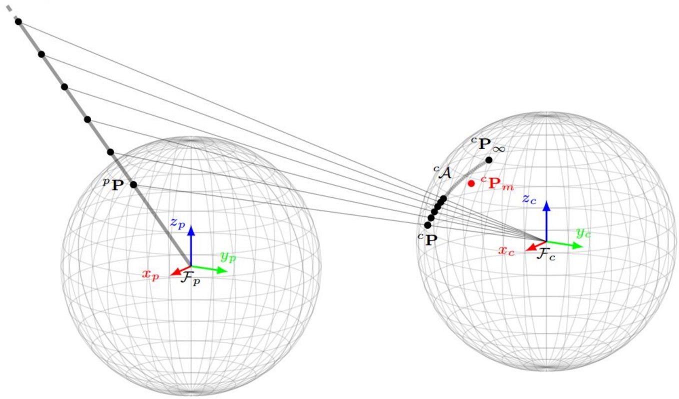
$$\mathbf{P}' = {}^{c}\mathbf{P}_{m} - \left({}^{c}\mathbf{P}_{m} \cdot \mathbf{n}\right)\mathbf{n}, \quad {}^{c}\mathbf{Q}_{m} = \frac{\mathbf{P}'}{|\mathbf{P}'|}, \qquad (3)$$



Object detection (cont.)

- If ${}^{c}\mathbf{Q}_m$ lies on ${}^{c}\mathcal{A}$ the distance of the point ${}^{c}\mathbf{P}_m$ to the arc ${}^{c}\mathcal{A}$ is calculated as $d({}^{c}\mathbf{P}_m, {}^{c}\mathbf{Q}_m)$
- If not, $\min\{d({}^{c}\mathbf{P}_m, {}^{c}\mathbf{P}), d({}^{c}\mathbf{P}_m, {}^{c}\mathbf{P}_\infty)\}$
- If the robot does not move or just rotate, then (4) is false, otherwise is true

$$({}^{c}\mathbf{P} \times {}^{c}\mathbf{Q}_m) \cdot ({}^{c}\mathbf{Q}_m \times {}^{c}\mathbf{P}_\infty) > 0 \quad \text{and} \qquad (4)$$
$$({}^{c}\mathbf{P} \times {}^{c}\mathbf{Q}_m) \cdot ({}^{c}\mathbf{P} \times {}^{c}\mathbf{P}_\infty) > 0.$$



[4] https://en.wikipedia.org/wiki/Bayes%27_theorem
[5] https://en.wikipedia.org/wiki/Von_Mises%E2%80%93Fisher_distribution

Object tracking on the sphere

- To pose a probabilistic model of the sensor measurement, they using Bayesian estimation [4] tracker based on the von Mises-Fisher distribution [5]
- The distribution has the following form, μ is the mean direction and κ is the concentration parameter

$$p(\mathbf{x}; \kappa, \boldsymbol{\mu}) = \frac{\kappa}{4\pi \sinh \kappa} \exp\left(\kappa \boldsymbol{\mu}^{\mathrm{T}} \mathbf{x}\right), \qquad (5)$$

- The larger the κ, the greater the clustering around the mean direction

Object tracking on the sphere (cont.)

- Two steps: (1)prediction and (2) update
- In prediction, the probability density function (pdf) is:

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})\, \mathrm{d}\mathbf{x}_{k-1}, \quad (6)$$

- In update, the result of update step is a VMF with the following parameters

$$\kappa_{ij} = A^{-1}(A(\kappa_i)A(\kappa_j)), \quad A(\kappa) = \frac{1}{\tanh\kappa} - \frac{1}{\kappa}. \quad (7)$$

$$\kappa_{ij} = \sqrt{\kappa_i^2 + \kappa_j^2 + 2\kappa_i\kappa_j(\boldsymbol{\mu}_i \cdot \boldsymbol{\mu}_j)}, \quad \boldsymbol{\mu}_{ij} = \frac{\kappa_i\boldsymbol{\mu}_i + \kappa_j\boldsymbol{\mu}_j}{\kappa_{ij}}. \quad (9)$$

- This will produce the estimate of the direction of the moving object
- They only track a single object, if there are multiple objects, it will track the closest measurement one in the update step

[6] https://en.wikipedia.org/wiki/Visual_servoing

Object following

- To follow the tracked moving object, they propose to use visual servoing technique to solve this problem
- Visual servoing [6]: a technique which uses feedback information from a vision sensor to control the motion of a robot
- They use cylindrical coordinate system, so the direction $\mathbf{x}_k$ on the sphere is:

$$\rho = \sqrt{s_x^2 + s_y^2}, \quad \theta = \arctan \frac{s_y}{s_x}. \qquad (10)$$

- Control law based on (1) linear velocity v and (2) angular velocity ω

$$\boldsymbol{v} = (v, \omega)$$

- Spatial velocity is $\dot{\boldsymbol{s}} = \mathbf{L}_s \boldsymbol{v}$

[7] http://ieeexplore.ieee.org/document/5067290/
[8] http://ieeexplore.ieee.org/document/5509199/

Object following (cont.)

- The interaction matrix is

$$\mathbf{L}_s = \begin{bmatrix} \frac{-\cos\theta}{P_z} & 0 \\ \frac{\sin\theta}{\rho P_z} & -1 \end{bmatrix}, \quad (11)$$

- Control law is

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = -\lambda \hat{\mathbf{L}}_s^{-1} \begin{bmatrix} \rho - \rho^* \\ \theta - \theta^* \end{bmatrix}, \quad (12)$$

$$\dot{\boldsymbol{s}} = \mathbf{L}_s \boldsymbol{v}$$

- e is the error (great cycle distance) of the control task

$$\lambda(e) = a \exp(-be) + c, \quad (13)$$

$a = \lambda(0) - \lambda(\infty), b = \lambda'(0)/a, c = \lambda(\infty)$
$\lambda(0) = 0.5, \lambda(\infty) = 0.05, \lambda'(0) = 0.5$

Experiment results

- Object moved so as to first distance itself from the desired position and then waited until the robot closed the distance by reducing the servoing task error to zero (demo)

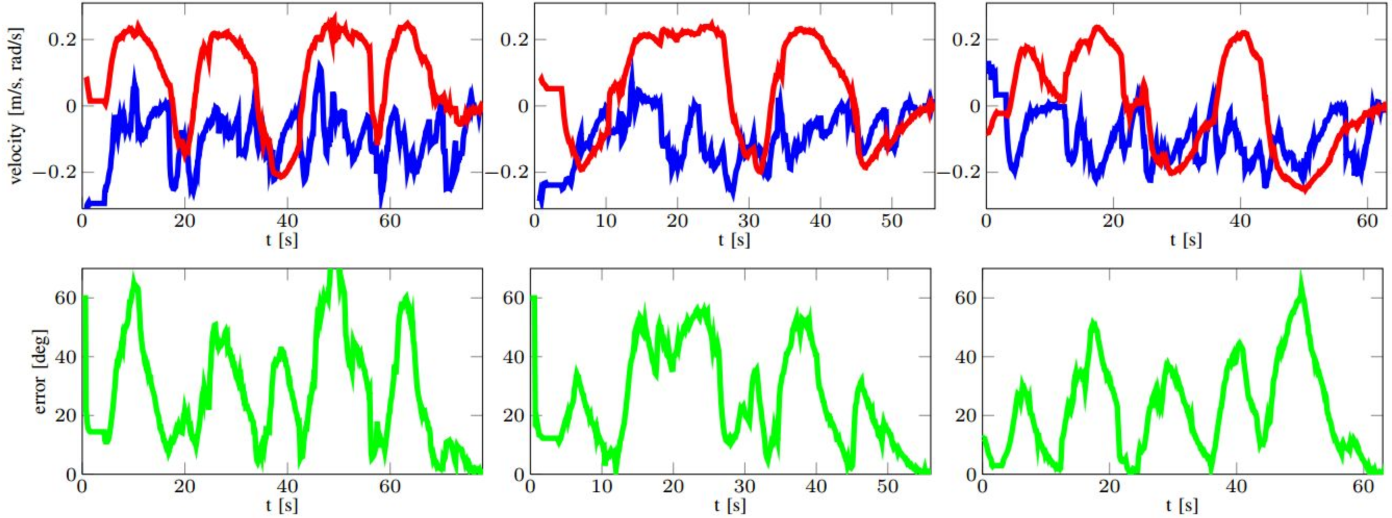


Fig. 5. Command velocities—linear (red) and angular (blue), and error of the control task (great circle distance from the desired to the estimated direction)

Conclusion

- They uses an omnidirectional camera, which contains a fisheye lens (a wide-angle lens) with a catadioptric lens (a mirror), mounted on a mobile robot to obtain the panoramic image
- The amount of information in a panoramic image increase the robot's abilities in reacting in the environment
- They present a method based on sphere for detecting moving object, tracking, and following it

Q & A