Available online at www.sciencedirect.com
ScienceDirect
Robotics and Autonomous Systems 55 (2007) 667–674
Robotics and Autonomous Systems
www.elsevier.com/locate/robot

Instantaneous robot self-localization and motion estimation with omnidirectional vision

Libor Spacek*, Christopher Burbridge

Department of Computer Science, University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, UK

Available online 24 May 2007

Abstract

This paper presents two related methods for autonomous visual guidance of robots: localization by trilateration, and interframe motion estimation. Both methods use coaxial omnidirectional stereopsis (omnistereo), which returns the range r to objects or guiding points detected in the images. The trilateration method achieves self-localization using r from the three nearest objects at known positions.

The interframe motion estimation is more general, being able to use any features in an unknown environment. The guiding points are detected automatically on the basis of their perceptual significance and thus they need not have either special markings or be placed at known locations.

The interframe motion estimation does not require previous motion history, making it well suited for detecting acceleration (in 20th of a second) and thus supporting dynamic models of robot's motion which will gain in importance when autonomous robots achieve useful speeds.

An initial estimate of the robot's rotation ω (the visual compass) is obtained from the angular optic flow in an omnidirectional image. A new noniterative optic flow method has been developed for this purpose. Adding ω to all observed (robot relative) bearings θ gives true bearings towards objects (relative to a fixed coordinate frame).

The rotation ω and the r, θ coordinates obtained at two frames for a single fixed point at unknown location are sufficient to estimate the translation of the robot. However, a large number of guiding points are typically detected and matched in most real images. Each such point provides a solution for the robot's translation. The solutions are combined by a robust clustering algorithm *Clumat* that reduces rotation and translation errors.

Simulator experiments are included for all the presented methods. Real images obtained from *ScitosG5* autonomously moving robot were used to test the interframe rotation and to show that the presented vision methods are applicable to real images in real robotics scenarios.

Keywords: Self-localization; Motion estimation; Omnidirectional vision; Omnistereo; Omniflow

1. Introduction

Amongst all robot sensors available, vision provides the most information, at the greatest range and often with the greatest accuracy.

Navigation and environment sensing are essential for autonomous mobile robots. The ability to quickly estimate position in an environment is often crucial. Omnidirectional vision offers detailed information about the entire surroundings and as such is ideally suited for use in robot localization.

Omnidirectional vision sensors have been constructed in many different ways. Tan et al. [13] use a pyramid of mirrors, and point multiple cameras at the pyramid. This configuration offers high resolution and the possibility of a single view point but is not isotropic and the registration and the physical arrangement of the cameras can be difficult. Rotating cameras and mirrors were used by Kang and Szeliski [6] and Ishiguro et al. [5]. However, difficulties were encountered with the registration and the motion delay of the camera. Wide angle and fish eye lenses have been used by Swaminathan and Nayar [12] and Shah and Aggarwal [9]. Satisfactory methods of autocalibration and rectification of the distortions of the true fish eye lenses have only recently been developed by Micusik and Pajdla [7].

Catadioptric omnidirectional sensors use a mirror and a camera. The mirror is rotationally symmetrical and the camera points at the mirror along its rotational axis. One

* Corresponding author.
E-mail address: spacl@essex.ac.uk (L. Spacek).

0921-8890/$ - see front matter
doi:10.1016/j.robot.2007.05.009

major advantage that this design has over some of the other omnidirectional sensors is that there is no need to wait for a moving camera or to synchronize multiple cameras. This means better suitability to dynamic environments and mobile robotics.

There are several types of mirror that can be used. We use the conical mirror. This has the benefit of producing no radial distortion or loss of radial resolution, as is produced by hyperbolic mirrors. However, conical mirrors produce multiple effective viewpoints, which was seen until recently as problematic. Spacek [11] shows that a single effective viewpoint is not actually required for a correct perspective projection and image unwarping.

We use a simulator capable of accurately modelling omnidirectional vision using ray-tracing techniques [2]. This allows us to accurately compare our results to the ground truth. We created a simulated robot containing two vertically aligned catadioptric omnidirectional sensors. The ray tracing methods reconstruct the images, as seen through the mirrors, including their inherent pixellation and resolution errors.

We first deploy omnistereo for range finding. Then we use the distance to the objects in a simple landmark-based localization method, using trialeration from the three closest objects at known positions.

We then develop an optic flow method to estimate the rotation of the robot and, using it, we show how to calculate the motion of the robot between two frames from only a single arbitrary fixed point at an unknown location.

Finally, we demonstrate the extensibility of the developed methods to real robotics with experiments that use a real robot set up in the same way as the original simulated robot.

2. Omnistereo range finding

The range to objects r can be calculated using a pair of omnidirectional images in much the same way as in classical forward-looking stereo vision. The formula for calculating the range [11] is:

$$r = \frac{vs}{h_{i1} - h_{i2}} - d, \tag{1}$$

where d is the distance from the camera lens to the tip of the mirror, s is the distance between the mirrors, v is the distance of the image behind the lens of the camera and $h_{i1} - h_{i2}$ is the radial disparity between the imaged object positions in the two images. See Fig. 1. The distance to an object can only be computed if the object lies in the common region. The common region for our mirrors has the vertical field of view of 45° and the horizontal field of view of 360°, which compares very favourably with classical stereopsis.

To calibrate v and convert between pixels and mm, v is calculated as in [11]:

$$v = \left(\frac{d}{R} + 1\right) r_m, \tag{2}$$

where R is the radius of the mirror, and r_m is the radius of the mirror in the image.

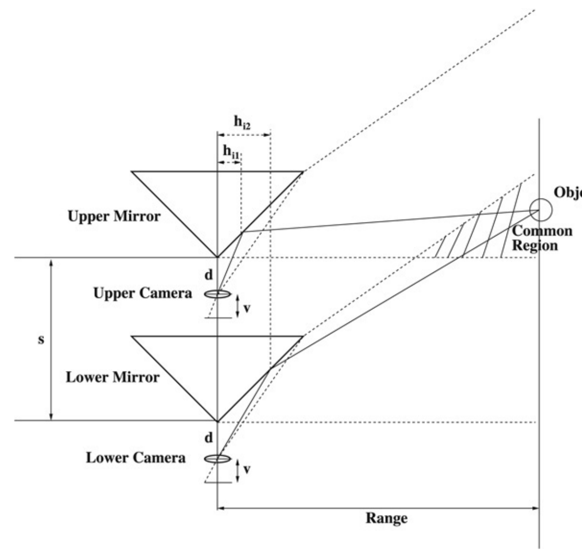

Fig. 1. Diagram showing the vertically aligned coaxial omnistereo configuration.

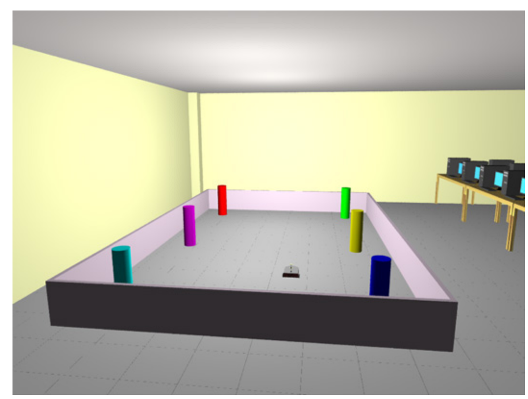

Fig. 2. The virtual lab. Six coloured pillars are positioned around the arena at known locations for landmark-based robot localization.

These formulae apply for conical mirrors with 90° angle at their apex. Other mirrors do not produce a cylindrical projection. However, similar formulae can be obtained for all circularly symmetrical mirrors. The registration issues connected with aligning the line of sight to the mirror axis are discussed in [11] and in [3].

2.1. Experiments

In our initial experiments we used the simulated arena of 6 m by 4 m as shown in Fig. 2. In all experiments reported in this paper the robots were configured for vertically aligned omnidirectional stereo as described here and shown in Fig. 1.

The robot was programmed with a random wandering, obstacle avoiding behaviour. This was run for 450 frames and all images were saved alongside the actual robot trajectory. These frames were then processed to calculate the distance to the pillars and the estimated position of the robot in the arena.

In order to achieve obstacle avoidance a map of the arena was programmed into the robot motion controller. The controller drives the robot forwards, checking the map until it is at a pillar or a wall. It then randomly turns either left or right