Bulletin of the *Transilvania* University of Braşov • Vol. 3 (52) - 2010
Series I: Engineering Sciences

COMPUTER VISION BASED MOBILE ROBOT NAVIGATION IN UNKNOWN ENVIRONMENTS

G. MĂCEŞANU[1] **F. MOLDOVEANU**[1]

Abstract: *There has been an increasing interest for mobile robotics structures because they allow making activities without human supervision. This paper presents the main problems of an autonomous mobile robotic platform, which uses digital image processing techniques for extracting important information from the environment. In the following, there will be presented acquisition approaches based on digital cameras, approaches that will be used for visual navigation. The navigation process is based on probabilistic methods. The simultaneous localization and mapping structures will be the one that integrates both the navigation process and the probabilistic approaches.*

Key words: *computer vision, image processing, autonomous mobile robot, navigation, simultaneous localization and mapping.*

1. Introduction

A large part of current robotics research and development is on aspects that make robot systems more autonomous and versatile. Robots have used vision for navigation to a greater or lesser extent for many years [5].

The earliest attempt at using computer vision was probably the robot named “Shakey” [15], robot built at Stanford University. It operated in an artificial environment with coloured blocks. It was called Shakey because of the way it vibrated when it moved.

Shakey was followed by the Stanford Cart [13] which used two cameras, one of which could be moved to provide different viewpoints for stereo vision. It was very slow and test runs took hours with the available computing power at the time. Image processing took so long that the shadows during outdoor test runs became a problem because they would actually move noticeably between images.

In the late 1980s, one of the best known robots was Polly, which was built by Ian Horswill [8]. This used a low resolution black and white camera to detect the floor and people. This robot had a built-in map of part of a building at MIT, including the place where the carpet changed colour so it could navigate across this apparent “boundary”. Polly was known to have several deficiencies, including stopping when it saw shafts of light coming in through the windows.

The ER-1 robot (Evolution Robotics 2005)

[1] Dept. of Automation, *Transilvania* University of Braşov.

used a web camera to perform what Evolution Robotics called visual simultaneous localization and mapping (vSLAM) [20]. This was basically place recognition based on building a large database of unique image features. These features were obtained using scale invariant feature transform algorithm (SIFT) [11]. It is common for SIFT-based systems to use several thousand features in an image.

Andrew Davison used features to create what he refers to as monoSLAM (SLAM with a single camera) [4]. This system built maps by tracking hundreds of features from one image to the next and eventually determining their physical location in the world by applying an extended Kalman filter (EKF).

In this paper, our goal is to survey the most important aspects of vision for mobile robot navigation. This paper is organized as follows. In Section 2, are presented the computer vision concepts. Main issues considered in the navigation of an autonomous robot platform will be presented in Section 3. Section 4 describes the most popular algorithms for vSLAM, then in the next section mapping and localization techniques are exposed.

2. Computer Vision

The primary objective of a computer vision system is to segment images to obtain useful information about the surrounding environment.

2.1. Digital Cameras

Digital cameras create images consisting of picture elements (pixels) there are quantization effects when a digital image is created.

The number of pixels across the image is known as the horizontal resolution. The vertical resolution is the number of rows or scanlines in the image. The ratio between the horizontal and vertical resolutions is the aspect ratio of the camera. Typical cameras have an aspect ratio of 4:3. The cameras used in computer vision vary considerably in their specifications. They have fairly low resolution which ensures that the amount of computation involved in processing images is not excessive.

Visual motion estimation techniques from monocular or stereovision sequences provide precise motion estimates between successive data acquisitions, but they are akin to dead reckoning [10].

Using two cameras with a reasonable distance between them allows the stereo disparity to be calculated, and hence depth or distance to objects in the field of view established. To extract depth information using stereo vision the correspondence between pixels in the two images must be established unambiguously [5]. This correspondence problem is a recurring theme in computer vision. It is often necessary to match points in two images.

2.2. Camera Features

A camera consists of an image plane and a lens which provides a transformation between object space and image space. This transformation cannot be described perfectly by a perspective transformation because of distortions which occur between points on the object and the location of the images of those points [7].

By taking pictures of a “calibration object” (usually a checkerboard pattern) from various different positions, it is possible to calculate the intrinsic parameters of the camera, which include the focal length and various distortion parameters. These parameters allow images to be undistorted by applying an appropriate algorithm [7].

Given the intrinsic parameters, it is also possible to calculate the extrinsic parameters for a given image using the same software. The extrinsic parameters consist of a rotation matrix and a translation vector that transform camera coordinates to real-world coordinates aligned with the calibration target. These parameters can be used to obtain the camera tilt angle.

One of the camera significant problems of many digital cameras is a limited field of view (FOV). A typical web camera has a FOV of 40°-60° (in contrast, humans have a FOV of up to 200°).

A restricted FOV means that the robot must "look around" to get a good view of the surrounding environment when building a map. Also, to make the best use of the available FOV, the cameras on robots are often tilted downwards as in Figure 1. Even with this tilt, there might be an area in front of the robot that cannot be seen, labelled as the blind area in Figure 1.

Panoramic cameras have been widely used to overcome FOV limitations [18]. There are various types of special-purpose cameras: omnidirectional cameras (360° FOV); panoramic cameras (around 180° FOV) and cameras with wide-angle or fish-eye lenses.

The common factor in all of these is significant distortion of the image. In these cases, standard perspective no longer applies and straight lines no longer appear straight in the image. Nayar [14] specifically developed a catadioptric camera (one that uses mirrors) so that he could extract true perspective images from the omnidirectional image.

2.3. Range Estimation

To build metric maps (maps that are drawn to scale) it is necessary to measure distances. Recovering depth information from images is one of the key areas of vision research. There are many different ways to do this.

If the pixels in the image can be classified as either floor or non-floor, this eliminates one degree of freedom in the 3D locations of the real-world points that correspond to the floor pixels, which must be on the ground plane (floor). Simple geometry can then be used to obtain range estimates assuming that the camera configuration is known [16].

Strongly related by range estimation are effects of camera resolution. Thus, consider for example one of the earliest digital cameras that only had a resolution of 64x48 pixels. Each horizontal scanline (one of the 48 rows of pixels) corresponds to a large distance across the floor in the real world. Now imagine a camera with a resolution of 640×480 (VGA resolution) that captures exactly the same scene. Obviously the higher resolution means that the distances to obstacles can be resolved more accurately.

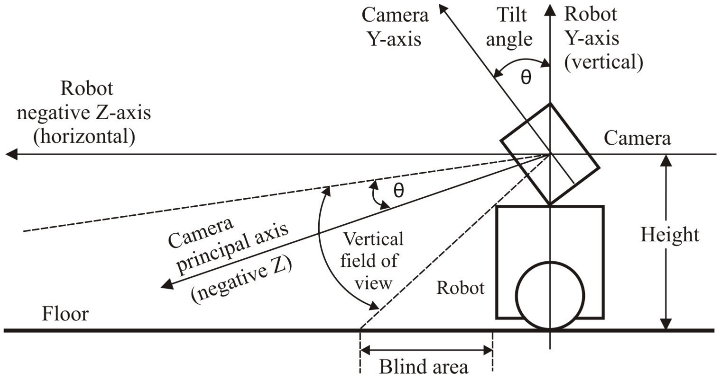

Fig. 1. *Camera orientation and vertical field of view*

Furthermore, the tilt angle of the camera has an effect as well. It is only necessary to consider two cases to illustrate this - a camera oriented parallel to the ground and one facing directly down at the ground.

For the horizontal camera, the centre scanline corresponds to the horizon, which is at infinity. One scanline below this is still a long way away, but significantly less than infinity, and so forth until the bottom scanline which is right in front of the robot.

On the other hand for a camera oriented vertically, the distance from the camera (measured across the floor) is zero at the centre scanline and only varies by some small amount from the top scanline to the bottom scanline of the image.

3. Navigation

The navigation could be described as the process accomplished by a mobile robotic platform, for determining the reasonable and safe path between a starting and a target point.

3.1. Visual Navigation

Robots have been navigating using vision since the days of the Stanford Cart [13]. Many robots today still use sensors such as sonar, infra-red or laser range finders, and do not have cameras at all. Even those that do have cameras often rely on them only to identify places or objects that they are searching for, but use sonar or lasers for actual navigation.

A wide range of image processing algorithms and image processing tools (the Intel Open Computer Vision Library) are available today as open source code. However, processing steps such as edge detection, segmentation, line detection etc. do not constitute vision on their own. Computer vision requires analysis of the scene and high-level understanding.

The most fundamental problem facing an autonomous robot using vision is obstacle detection. However, despite decades of research, there is no agreement within the computer vision community on how best to achieve this task, especially as a precise definition of obstacle detection is surprisingly difficult [20].

In its simplest form, obstacle detection is the process of distinguishing an obstacle from the floor. It is not necessary to understand what is seen in order to avoid obstacles - simply distinguishing between the floor and all other objects (even moving humans) is sufficient [20].

3.2. Image Segmentation

One approach to image segmentation is to use clustering whereby pixels that are similar (according to some measure of similarity) are grouped together. A popular clustering algorithm is k-means [6].

Ulrich and Nourbakhsh [17] developed a system that was intended to be used in a variety of environments. The simplified version of their method consists of the following four steps [2]:

1. filter colour input image;
2. transformation into hue saturation intensity colour space;
3. histogramming of reference area;
4. comparison with reference histograms.

The primary objective was to relax the constraint that the reference area must be free of obstacles. The reference area was only assumed to be the ground if the robot had just travelled through it. A “reference queue” was kept to “remember” the past reference areas [17].

Detecting the floor has also been a common topic in the Robot Soccer literature. James Bruce [1] presented a scheme using the YUV (YUV is a colour space typically used as part of a colour image pipeline) colour space that allowed them to track several hundred regions of 32 distinct colours at a resolution of 640 × 480 at a frame rate of 30 Hz using a 700 MHz PC.

Quite recently, the winning team in the DARPA Grand Challenge in 2005 used a mixture of Gaussians to extract the road surface from camera images [3].

Other problem which appears in image processing is about object tracking. Difficulties in tracking objects can arise due to abrupt object motion, changing appearance patterns of both the object and the scene, no rigid object structures, object-to-object and object-to-scene occlusions, and camera motion. Tracking is usually performed in the context of higher-level applications that require the location and/or shape of the object in every frame. Typically, assumptions are made to constrain the tracking problem in the context of a particular application [19].

4. Probabilistic Filters

Most modern SLAM algorithms rely on the use of probabilistic filters. The most commonly used forms of filters are Kalman filters [16] and particle filters [12].

The reason for the popularity of probabilistic techniques stems from the fact that robot mapping is characterized by uncertainty and sensor noise.

Probabilistic algorithms approach the problem by explicitly modelling different sources of noise and their effects on the measurements [16].

4.1. Kalman Filters

A Kalman filter is basically a recursive state estimator [16]. The "state" consists of a set of "features" observed in the environment plus the robot's pose. Estimating the location of the features effectively builds a map.

The reason for the early popularity the Kalman filter is that it is analytically tractable and the update equations can be expressed in closed form. Being a recursive filter, it is also computationally efficient [2].

Kalman filters approximate the robot motion model using a linear function obtained via Taylor series expansion. The resulting Kalman filter is known as EKF, and single motion commands are often approximated by a series of much smaller motion segments, to account for nonlinearities [16].

4.2. Particle Filters

A popular form of localization is called Monte Carlo localization (MCL) [12].

MCL represents the belief in the current pose of the robot as a set of particles. In simple terms, a particle consists of a pose estimate and any necessary state information that is required to perform updates to the pose. As the robot moves, these particles trace out trajectories (or paths) on the map [12].

If the set of particles is sufficiently large and has an appropriate distribution that adequately represents the true posterior probability distribution, then the average of the particles should be near to the actual pose of the robot. Ideally, one of the particles should be very close to correct. If there is no such particle, then there are insufficient particles in the set and the filter is likely to diverge.

5. Mapping and Localization

The process of SLAM means tracking the position of a mobile robot relative to its environment and building a map of the environment [2]. This has been a central research topic in mobile robotics. Accurate localization is a prerequisite for building a good map, and having an accurate map is essential for good localization. Therefore, SLAM building is a critical underlying factor for successful mobile robot navigation in a large environment, irrespective of what the high-level goals or applications are [2].

5.1. Mapping

One obvious function of a map is to enable the robot to remember the terrain that it has driven over. This provides positive proof of free space. The robot might use this prior information (in its memory) to predict what a new view should look like for comparison with actual sensor information. It can adjust its pose in the previous "obstacle map" according to its own motion and then compare what it sees with what is already in the map [16].

Before mapping can be done using vision, it is necessary to transform the camera images back to representations of the real world.

To draw maps correctly, the robot must always know precisely where it is. In simulation this is not a problem because the robot's motions are always perfect. However, in the real world the robot's pose can only be estimated because there are random errors in motions. Determining the robot's pose is a localization problem. As should already be apparent, in practice it is not possible to completely separate localization from mapping [2].

There are two fundamentally different types of maps: topological and metric.

The topological map, extracts the environmental entities as abstracted models like nodes and edges. Then, it represents the environment as spatial relations of those entities, usually in graph structure. This graph representation is useful to generate a robot path as a sequence of places to a goal position [9].

The metric map represents exact location of geometric entities with respect to a reference frame. This exact representation is helpful for the robot to perform elaborate tasks which require high accuracy [9].

5.2. Localization

Localization is the process that a robot uses to determine where it is in the world. Usually this is done by comparing surrounding features with a map. The definition of localization can therefore be re-stated as the process of updating the pose of a robot based on sensor input [2].

Sonar sensors were used in a lot of the early research on localization. Typically the sensors were arranged in a ring around the perimeter of the robot. This meant that a full 360° sweep could be obtained without moving the autonomous mobile robot.

Localization methods fall into two broad categories [2]:

1. Kalman filters which track features, using normal distributions to estimate errors in feature locations.
2. Markov localization methods, also called MCL or particle filters, which approximate an arbitrary probability distribution using a set of particles and operate on a grid-based map.

It is important to note that most popular localization methods make the assumption that the system can be represented as a first-order Markov process. In terms of localization, this means that the robot's pose at a particular time *t*, depends only on its pose at the previous time step, (*t*–1), and the control input applied to the robot's motors in the intervening time interval. In general, the pose at time *t* might depend on a finite number of previous poses, but this is ignored [16].

5.3. Simultaneous Localization and Mapping

For autonomous mobile robots to operate in human environments they need SLAM, also referred to as concurrent mapping and localization (CML).

The primary problems that SLAM has to solve are:

1. determining the robot's pose from uncertain input data;
2. estimating the relative position of observed features from measurements containing noise.

Building a map is not one of the significant problems if the input data is accurate. Also, the exploration algorithm does not have a strong effect.

However, if the robot's pose is not correct, or the observations contain gross errors, then features will be inserted into the map in the wrong locations. These errors accumulate, resulting in distortions of the map. It is even possible for the robot to become completely lost [16].

Digital cameras are the most recent type of sensors to be used and there is a great deal of research currently in progress on using vision for SLAM. One of the key advantages of vision, which was part of the initial impetus for this research, is that cameras are now very cheap. The disadvantage is that digital cameras are very complex sensors [2].

5.4. Visual Simultaneous Localization and Mapping

Visual localization and mapping for mobile robots has been achieved with a large variety of methods. Among them, topological navigation using vision has the advantage of offering a scalable representation, and of relying on a common and affordable sensor [2].

There are several research threads running through the field of vSLAM, including visual localization and visual odometry. Also, some researchers have attempted to reconstruct 3D maps, whereas others assumed that the robot moved in 2D on a ground plane and used the homography of the ground plane as a constraint [16].

6. Conclusions

This paper has outlined a wide range of different subject areas that must be addressed in order to perform mapping using vision. From the wide range of SLAM approaches, the vSLAM is the newest and most challenging one. In this type of SLAM the classic sensors were replaced with a new type of sensor, the video camera. Currently, in the scientific community, there is a lot of work on mapping using vision based robotic platforms that are able to produce maps of unknown environments and at the same time to self localize using the created map.

Acknowledgements

This paper is supported by the Sectoral Operational Programme Human Resources Development (SOP HRD), financed from the European Social Fund and by the Romanian Government under the contract number POSDRU/88/1.5/S/59321.

References

1. Bruce, J., Balch, T., et al.: *Fast and Inexpensive Color Image Segmentation for Interactive Robots.* In: Proceedings of the International Conference on Intelligent Robots and Systems, Takamatsu, Japan, October 2000, p. 2061-2066.
2. Choset, H., Lynch, K., et al.: *Principles of Robot Motion-Theory, Algorithms, and Implementation.* London, England. The MIT Press, 2005.
3. Dahlkamp, H., Kaehler, A., et al.: *Self-Supervised Monocular Road Detection in Desert Terrain.* In: Robotics: Science and Systems, Philadelphia, USA, 2006, p. 197-212.
4. Davison, A.J., Murray, D.W.: *Mobile Robot Localisation Using Active Vision.* In: Proceedings of the 5th European Conference on Computer Vision, Freiburg, Germany, 1998, p. 809-825.
5. DeSouza, G.N., Kak, A.C.: *Vision for Mobile Robot Navigation: A Survey.* In: IEEE Transactions on Pattern

Analysis and Machine Intelligence **24** (2002), p. 237-267.

6. Duda, R.O., Hart, P.E., et al.: *Pattern Classification.* 2nd Ed. New York. Wiley, 2001.
7. Gonzalez, R.C., Woods, R.E., et al.: *Digital Image Processing Using MATLAB*. Pearson Prentice Hall, 2004.
8. Horswill, I.D.: *Polly: A Vision-Based Artificial Agent.* In: Proceedings of the Eleventh National Conference on Artificial Intelligence, Washington, USA, 1993, p. 824-829.
9. Jinwoo, C., Minyong, C., et al.: *Topological Modeling and Classification in Home Environment Using Sonar Gridmap.* In: IEEE International Conference on Robotics and Automation, Kobe, Japan, 2009, p. 3892-3898.
10. Lemaire, T., Berger, C., et al.: *Vision-Based SLAM: Stereo and Monocular Approaches.* In: International Journal of Computer Vision **74** (2007), p. 343-364.
11. Lowe, D.G.: *Distinctive Image Features from Scale-Invariant Keypoints.* In: International Journal of Computer Vision **60** (2004), p. 91-110.
12. Montemerlo, M., Thrun, S., et al.: *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem.* In: Proceedings of the National Conference on Artificial Intelligence, Edmonton, Canada, May 2002, p. 593-598.
13. Moravec, H.P.: *Towards Automatic Visual Obstacle Avoidance.* In: Proceedings of the 5th International Joint Conference on Artificial Intelligence, Cambridge, UK, 1977, p. 584-587.
14. Nayar, S.K.: *Catadioptric Omnidirectional Camera.* In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Puerto Rico, May 1997, p. 482-488.
15. Nilsson, N.J.: *Shakey the Robot.* In: Technical Report Nr. 323, SRI International, Menlo Park, USA, 1984.
16. Thrun, S.: *Robotic Mapping: A Survey.* In: Exploring Artificial Intelligence in the New Millenium, Technical Report, Pittsburgh, USA, 2003, p. 1-35.
17. Ulrich, I., Nourbakhsh, I.: *Appearance-Based Place Recognition for Topological Localization.* In: IEEE International Conference on Robotics and Automation, San Francisco, USA, 2000, p. 1023-1029.
18. Yamazawa, K., Yagi, Y., et al.: *Obstacle Detection with Omnidirectional Image Sensor HyperOmni Vision.* In: IEEE International Conference on Robotics and Automation **1** (1995), Nagoya, Japan, p. 1062-1067.
19. Yilmaz, A., Javed, O., et al.: *Object Tracking: A Survey.* In: ACM Computing Surveys **38** (2006), p. 67-112.
20. Zhang, Z., Weiss, R., et al.: *Obstacle Detection Based on Qualitative and Quantitative 3D Reconstruction.* In: IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997), p. 15-26.
21. *** *Robots, robots kits, OEM solutions: Evolution Robotics.* Available at: http://www.evolution.com/er1. Accessed: 10-12-2009.