Mapping with Monocular Vision in Two Dimensions

Nicolau Leal Werneck, Universidade de São Paulo, Brazil

Anna Helena Reali Costa, Universidade de São Paulo, Brazil

ABSTRACT

This article presents the problem of building bi-dimensional maps of environments when the sensor available is a camera used to detect edges crossing a single line of pixels and motion is restricted to a straight line along the optical axis. The position over time must be provided or assumed. Mapping algorithms for these conditions can be built with the landmark parameters estimated from sets of matched detection from multiple images. This article shows how maps that are correctly up to scale can be built without knowledge of the camera intrinsic parameters or speed during uniform motion, and how performing an inverse parameterization of the image coordinates turns the mapping problem into the fitting of line segments to a group of points. The resulting technique is a simplified form of visual SLAM that can be better suited for applications such as obstacle detection in mobile robots.

Keywords: Computer Vision, Data Processing, Mobile Robotics, Mobile Technologies, Monocular SLAM

INTRODUCTION

Estimating the localization of a camera and a map of an environment at the same time is a problem that has been approached by researchers from two directions (Strasdat et al., 2010). In Computer Vision it is the *Structure from Motion* (SFM) problem, and in Mobile Robotics it is a case of *Simultaneous Localization and Mapping* (SLAM) where the sensor is a camera. Most of the existing techniques rely on some sort of image feature extraction to produce landmark observations that are further analyzed to estimate the camera track and landmark positions. This article demonstrates how simple image processing methods, similar to what is used in robot navigation and localization, can be used to perform environment mapping. The proposal takes in consideration some restrictions in the environment and motion that lead to simpler processes in the whole mapping system, from the image analysis to the data association and parameter estimation. Some performed tests demonstrate the possibility of applying this technique for obstacle avoidance based on monocular mapping, and also reveal the necessary steps for the development of a more elaborate system for 3D reconstruction of indoor environments.

This research concerns images taken from indoor and other man-made environments, from where large edges can be extracted. The

localization of robots in bi-dimensional maps under these conditions is a well-studied problem (Borenstein, 1996). The landmarks used are vertical edges, and their positions on the ground plane constitute the map. While localization has been well studied, the problem of building maps using only cameras under these or other restrictions, but without the help of metric sensors, has only recently received more attention. Many researchers studied systems where a stereoscopic apparatus perceives visual landmarks with three-dimensional location and corresponding visual descriptors to perform visual SLAM (Sünderhauf & Protzel, 2007). These landmarks are usually detected using punctual feature extraction algorithms, and the stereoscopic rig is used as a metric sensor.

Stereo rigs are inherently wide, hard to build, and have limited precision. These problems motivate the research on monocular SLAM techniques that have been growing lately. Some recent developments in this area have been the use of edge landmarks (Eade & Drummond, 2009), the use of alternative parameterizations (Solá, 2010) and also the creation of maps with non-calibrated cameras that are correct up to scale parameters (Civera et al., 2007). The use of edge landmarks is very important to many application scenarios, especially indoor environments, and the alternative parameterizations make the problem more suited to filtering techniques such as the extended Kalman filter. The creation of scaled or otherwise transformed maps also makes the problem easier to approach, and makes the technique more generally applicable.

The research presented in this article relates to these recent trends. The features extracted from the input images are intended to be edges, and they are detected by a search for peaks in the image gradient over a line orthogonal to their direction. This kind of detection is very simple but has great application potential, and was inspired from an idea by Nourbakhsh (1997). One of its great advantages is making use of information that is frequently ignored in systems that only work with features that have low auto-correlation in all directions, which excludes edges.

This form of image analysis is not uncommon in mobile robotics, but it has been usually limited to detecting obstacles at ground level by the segmentation of the ground in the lower portion of images (Howard & Kitchen, 1997; Ulrich & Nourbakhsh, 2000; Lenser & Veloso, 2003). These techniques are not able to identify when the detected borders are the edges of walls, drop-offs or just drawings on the ground. They also can't handle correctly suspended obstacles such as hanging signs, railings, tabletops and other furniture, and even underestimate their proximity. Our technique applied to the analysis of horizontal edges crossing the central column of input images overcomes these difficulties by measuring the distance and elevation of the features ahead, using measurements from multiple images.

The objective is to move away from the assumption that the detected points are at ground level, and perform something closer to more general mapping techniques. But the creation of precise maps is only a secondary goal, the technique is still useful without careful measurements of camera parameters such as its focal length, or also its actual position or speed. The consequence is that the maps produced are accurate only up to two unknown scale factors, but still useful for tasks such as obstacle detection.

The next sections present a description of the geometry of the problem and of the necessary image processing steps. A technique to detect obstacles in mobiles robots is proposed, and there is also an analysis of detected evidences taken from an experiment with a video recorded at an indoor environment with a conventional camcorder. Ways to produce long-term maps from the evidences are finally suggested for future development.

PROBLEM GEOMETRY

Given two camera positions and orientations, and a pair of corresponding points projected

over the image plane of each camera it is possible to calculate the position of the projected point in space, as long as the point does not lie in the line over the two optical centers from the cameras (Trucco & Verri, 1998). If a number of images from a certain environment are available, and the positions of the camera for each image is known, and the image coordinates of a certain point is known in two or more images, it is possible to determine the location of the point in space. In the context of our problem the points in space are the landmarks, and their projections detected in the images are the evidences. To build a map is to calculate the space coordinates of the landmarks from the input image coordinates and camera positions.

More general image-based localization and mapping problems consider complete 6 degrees of freedom camera movements in a three-dimensional environment, where landmarks are three-dimensional points. In our case space is restricted to two dimensions. The points that are mapped are actually, most of the time, the loci where lines in three dimensions cross the plane defined by the focal point of the camera and the line over the image that is selected for analysis. The camera moves over this same plane where the landmarks are located, and their projections over the image planes are determined by only one coordinate: the location over the extracted line. A further restriction is that the camera moves along the optical axis.

Figure 1 displays how these entities relate to each other. Our problem consists in determining the coordinates $\left(x_j, y_j\right)$ of each landmark j given the image coordinates z_{jk} of the projec-

tion of the landmark detected over the image plane of a camera located at position p_k. The camera has a focal distance of f, and the intrinsic parameters related to sensor size and resolution are not considered because they can all be modeled by a virtual f value that accounts for them.

The formula for the image coordinate z_{jk} can be deduced by triangle similarities, and is:

$$z_{jk} = \frac{y_j f}{x_j - p_k} \tag{1}$$

Given the corresponding evidences from landmark j at images a and b, the formulas for the landmark coordinates are:

$$x_j = \frac{p_b z_{jb} - p_a z_{ja}}{z_{jb} - z_{ja}} \tag{2}$$

$$y_j = \frac{p_b - p_a}{f} \cdot \frac{z_{jb} z_{ja}}{z_{jb} - z_{ja}} \tag{3}$$

Equations number 2 and 3 constitute the basis of all the mapping techniques presented in this article. They allow the construction of a map from a set of images obtained by moving a camera in the direction of its projection axis, provided that the position of the camera at each frame and the correspondence between the points on the different images are known. To create a coherent map taking into account more than two images is not simple, and dif-

Figure 1. Perspective view of three edges projected into one image plane (left), and relations between one landmark and two image planes, seen from above (right)
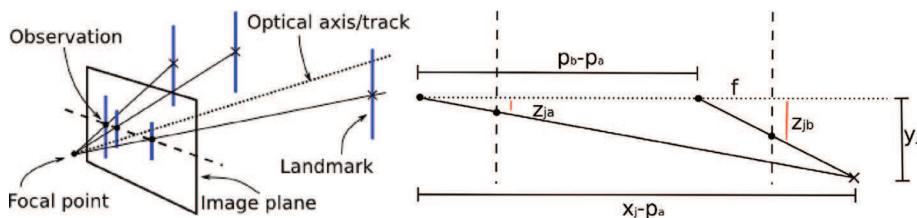
ferent estimation techniques can be used for that. But given any two images an *ad hoc* map can be calculated containing the positions of the landmarks visible on both of them, even though noise and uncertainties may cause map changes from one pair of images to another.

One first interesting characteristic of these formulas is that x_j does not depend on f, but only on the disparity between the image coordinates $z_{jb} - z_{ja}$ and on the difference of the products of the evidence parameters. On the other hand y depends on f, on the disparity and also directly on the difference of the locations $p_b - p_a$.

Another interesting fact from these formulas is that z_{jk} as a function of p_k describes a hyperbole on space, so the inverse of z_{jk} can be defined as a new value w_{jk} that is a linear function of p_k. If the camera moves at constant speed, that allows the use of integer values for p_k, taken from the index of each image on the sequence. A multiplicative constant s can also be introduced to model the camera speed and frame rate. The formula for w_{jk} thus becomes:

$$w_i = \frac{x_j}{y_j f} - \frac{s}{y_j f} \cdot p_i = u_j - v_j p_i \qquad (4)$$

This formula means that, as the camera moves ahead in constant speed and direction, any landmark j produces a sequence of image coordinates z_{jk} whose inverse values w_{jk} are decremented at each frame by a constant value v_j. Given a set of values w_{jk} and p_k we can calculate u_j and v_j and determine the values of x_j and y_j up to the scaling factors f, related to the intrinsic parameters of the camera, and s related to the speed of movement and to the frame rate of the camera.

OBSTACLE DETECTION

This section describes how to apply the presented formulas to detect obstacles for mobile robot navigation, although other applications are possible. An initial image processing stage produces a set of evidences, which are the location of matching points in pairs of images obtained during camera movement. These matching points are then used to create a local map containing the distance and height of the objects ahead, and from the location of these points it is possible to infer that obstacles may exist ahead of the robot, specially suspended and negative obstacles.

Image Processing

The image processing stage of the proposed obstacle detection system begins with the acquisition of a pair of images from a camera fixed on a robot while it moves over a straight path. A sequence of images is analyzed by providing successive image pairs as inputs. If necessary, frames can be ignored (dropped) in order to obtain pairs of images taken from more separated positions. Once the image pair is available, the following steps are taken.

First, each input image can be filtered to reduce the noise and textures, but with care to preserve the edges. Some possible filters to be applied in this step are Gaussian blurs, moving averages, median filters and even non-linear diffusion filters. Since only a strip of the image is analyzed, techniques that are usually considered very demanding in computational terms might be appropriate. It is also possible to apply a more intense smoothing on the direction of the edges that are being searched for. At the end of this step a single string of pixels is extracted from the image to be used in the next steps. For the obstacle detection application the edges are supposedly horizontal and the central column of the image is taken.

The next step is to locate the positions of the landmark projections on the image. The

extracted string of pixels is first filtered again to produce the derivative at each channel, and then the peaks of this derivative are located. This assumes that landmarks are junctions where objects of different colors meet. Colors should not vary much within object surfaces, but significantly between different objects. Multiple techniques can be used for finding the derivative peaks, one example is to calculate the successive differences and then sum the squared values of the derivatives of each channel. The local maxima of this sequence that are larger than a certain threshold are selected as the locations of the projections.

Given the assumed image coordinates of the landmarks projections calculated on the second step, the third step is to match the projections over image pairs. One possibility is to perform the previous detection only in one of the images and then search for the corresponding locations on the other image using a similarity function based on the pixel intensity values. For example, one can extract from the first image windows of a few pixels around the detected locations, and then compare them to patches of the same size from the second image. The patch that is elected as the matching position on the second image is the one that scores better on a similarity test such as the *squared sum of differences* (SSD) (Sünderhauf & Protzel, 2007; Barra et al., 2009).

This search can be done on just a single direction, since objects move away from the center as the camera moves ahead, and the sizes of the windows and the search regions depend on the application. Other possibilities are using a more accurate phase correlation search, or locating the points in the second image with the peak searching technique and then choose the best match comparing the pixel intensities. Regardless of this choice, at the end of this third step we have taken a pair of images and found matching image coordinates of landmark projections.

Map Creation

After the pairs of matching image coordinates are calculated Equations 2 and 3 can then be used to find the landmark locations in 3D-space (x_j and y_j). If only a local map is desired, with the robot in the origin of the reference frame, we have $p_b = 0$ and $p_a = -s$, and the formulas become Equations 5 and 6:

$$x_j = s \cdot \frac{z_{ja}}{z_{jb} - z_{ja}} \tag{5}$$

$$y_j = \frac{s}{f} \cdot \frac{z_{jb} z_{ja}}{z_{jb} - z_{ja}} \tag{6}$$

Figure 2 shows the landmarks detected on a sequence of images produced with a computer graphics program (Roosendaal, 2004). The images simulate what a small robot moving in uniform motion towards the edge of a table in a living room should see. At the right of the figure there are two sample frames from the sequence, showing how a carpet and also the baseboard over the joint between the wall and the floor are initially occluded by the table but eventually become visible.

The two graphics at the left of Figure 2 show the inverse of the image coordinates detected by both methods described previously, one graphic for negative coordinate values (upper half of image, left if transposed) and the other for positive. The vertical axis of the two graphics is the index of each frame in the video. This index provides the p_k values used to apply Equations 5 and 6. The points are the detected location of the landmark projections in each image.

The circles were detected with the first method, and are the peaks of the derivative of the pixel intensities. The crosses were found by searching for matches of these detected points on the next image in the sequence using the SSD score. The matching technique applied was very simplistic, and as a consequence only a few of the crosses are precisely on top of corresponding points in the same frames. But this result is good enough for a simple obstacle detection system.

The values of the detected points were normalized so the image edges are the -1.0 and

Figure 2. Experiment with a CGI simulation of a robot approaching a desk edge. The left graphics show how the w_{jk} values decrease linearly with time drawing lines of different parameters for each landmark. The frames at the right show the initial occlusion of the carpet and baseboard. They produce lines that stem from the table one, accordingly. The leftmost graphic shows the approaching of the edge between the wall and the ceiling.
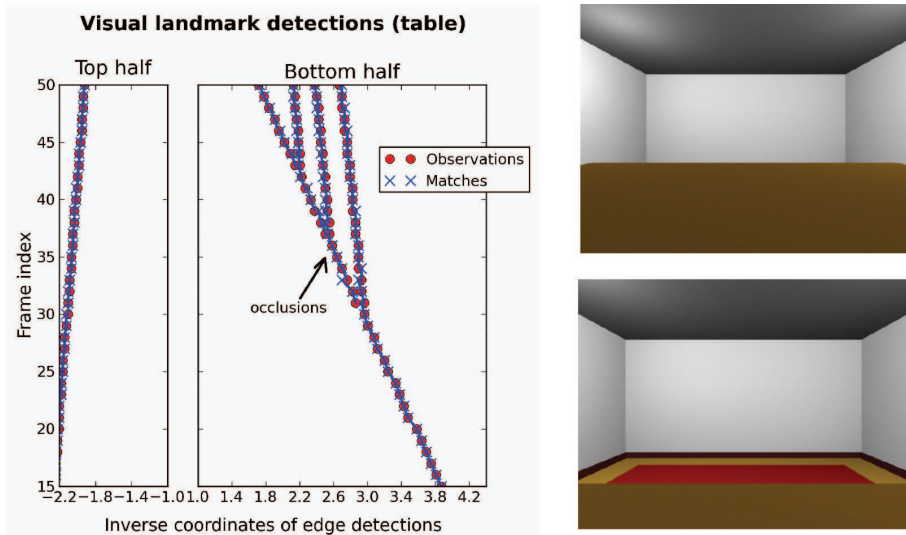
1.0 coordinates, and then inverted to produce the w_{jk} values used in the graphics. Because of the normalization, all of the w_{jk} have absolute values larger than 1. The left side of the image has the negative values, and the origin is the center of the picture, assumed to be the projection center. As we move away from the origin in the horizontal axis of the graphics, we start looking at landmarks projected closer to the center of the image. The detection of these landmarks is too noisy, so the plotted range refers to just part of the way from the edges to the center of the image.

The detected line inclinations v_j for the further and lower landmarks, related to the baseboard and the carpet, are lower than the inclination of the curve related to the table edge, indicating that this landmark is potentially the edge of a negative obstacle. The initial occlusion of the further landmarks is another evidence of that, all supposing the floor is neither trans-

lucent or reflexive. This identification of dropoffs from the detection of non-occluded low features on the environment is related to Murarka et al. (2009).

EXPERIMENT WITH REAL IMAGES

This section describes the analysis of a video recorded with a portable device to evaluate the applicability of the proposed technique in this practical case, and show the most relevant imperfections that should be considered when using this kind of data source. The camera used was a standard miniDV consumer camcorder, a Samsung SC-D391. The video was recorded in the camera as it was carried down a hallway pointing roughly towards the vanishing point, and analyzed afterward. To avoid the excessive shaking of the camera it was tied to a stabilization mount, similar to what is widely used by professionals of cinema and television.

Detected Features

Figure 3 displays two frames from the recording, and two graphics with the normalized inverse of the detected coordinates, like in Figure 2. The camera was pointing approximately to the direction of movement, and closer to the right wall than to the left. The section displayed in the graphics was recorded over a 15 meters movement. The dots in the graphics are the observations, with different colors and symbols for detections with positive and negative gradient value on the green channel. The lines are the result of a simplistic optimization procedure to fit landmarks parameters to the observations. Each of these landmarks were estimated individually from initial estimates chosen by hand, and the optimization found points that are local minima of a loss function that is the sum of absolute prediction errors considering all observations, limited to 10 pixels. This is similar to function used in, for example, MLE-SAC (Choi et al., 2009).

Despite the oscillations due to the shaking of the camera it is possible to detect some interesting features on the graphic at the left of Figure 3. There is a moment when a landmark occluded by a column becomes visible at ($p=130, w=1.6$), and also a region where objects inside a room can be seen through an open door, near ($p=240, w=1.1$). The curves on the left side of the graphics are also less steep as a function of p_k, indicating that these landmarks are further from the camera track.

In spite of the oscillations at the graphics of Figure 3, the result points to a way to create maps from environments given a recording like the one done for this experiment. Once all detections are obtained, the problem of creating a map becomes the identification of the parameters of the lines that interpolate the points. The matching of the detections between the images becomes the problem of determining to what line each point belongs, and the calculation of the landmark coordinates is the determination of the best parameters for the lines. To solve

Figure 3. Experiment with images recorded with a camcorder. The two frames at the right are samples taken from the recording. The curves in the graphics at the left are not straight as in the simulation because of imperfections such as slow changes in the orientation of the camera and accelerations. It is nonetheless possible to identify individual curves drawn by each landmark, notice occlusions by walls and verify the relative positions of the landmarks.
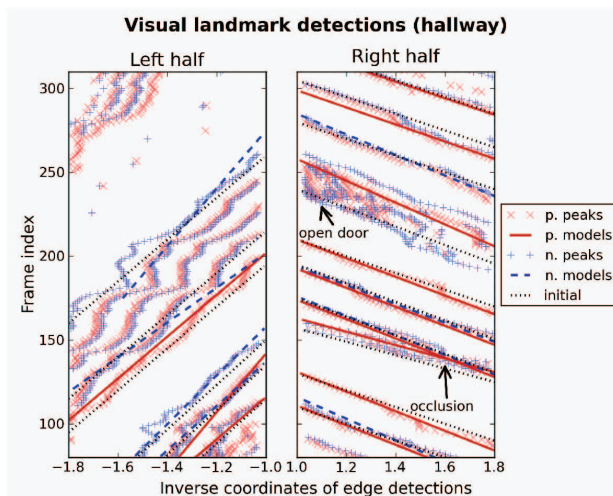
this problem a system designer might apply techniques such as the Hough transform, k-means, RANSAC (Choi et al., 2009), or the expectation-maximization algorithm. (Nguyen et al., 2007)

Sensitivity of Image Coordinates to Imperfections

The theory developed here assumes many conditions that while reasonable and not uncommon, are hard to attain with perfection. And while in robots or other fast moving vehicles the results might be satisfactory, this is definitely not the case for portable devices being carried by people. But even so Figure 3 displays something very close to what it was expected from our theory, the landmark projections formed approximately straight lines on the graphic.

An analysis of the sensitivity of the technique to the possible sources of imperfections can show why the results looked that way, and also point at how one should proceed to try to detect the sources of error and eliminate them. The main conditions are that the camera must move at a constant speed and orientation in the direction of the optical axis while detecting along the way the bearings of edges orthogonal to the path. Let's consider what happens when each of these conditions is not met:

Variable speed: If the camera accelerates the factor *s* is increased, and the inclination v_j of the line relative to each landmark also increases, making the curves more horizontal. When the speed decreases the curves become more vertical, and a full stop would produce perfectly vertical curves, because the landmark positions would not change with time.

Imperfect orientation: Our technique involves analyzing a single line on the image plane. This line and the focal point of the camera define a plane in the three-dimensional space. The measured evidences are the points where this plane crosses the edges on space that are orthogonal both to the motion direction and this plane. Because the landmarks are long straight lines, it makes little difference whether the plane tilts a little. What happens is just a scaling of the coordinates of the landmarks, but they are still detected.

Suppose a corridor with vertical lines and a camera moving along its center. If the image plane rotates around the horizontal axis the estimated x_j will be scaled up, and the landmarks will appear to be more distant. If the camera rolls, rotating around the optical axis, the y_j will be scaled. Moving the camera up, while keeping orientation, doesn't change anything, and moving to the sides only translates the y_j. It is also important to notice that picking up an image line that does not go exactly through the projection center has the same effects of rotating the camera in these axes, but the value of *f* is affected directly. It is only the rotation of the camera around the vertical axis that offers greater challenges, and should be specially avoided.

Imperfect edges on space: If the edges on space are not perfect straight lines orthogonal to the motion direction, and the camera is moving perfectly, then there is no problem. The only problem in this case is that curved surfaces on the environment will not be properly detected, their w_{jk} will generate curves instead of straight lines. But if the camera starts shaking negative effects can be perceived. First of all, tilted lines will not have their landmark locations scaled the same way as straight lines. Another important issue is that short lines will be detected or not as the projection plane crosses them or not.

In practice all lines are limited, and if the optical axis is not aligned to the motion direction a system designer might consider not taking the central line or column for the analysis, but instead searching for a line that

defines a plane that is more normal to the environment edges. This alignment will provide more and better detections, and the only effect is a scaling of the landmarks parameters. The principal point of the image is less strategic as a starting point for searching edges than the vanishing points or the epipoles (Trucco & Verri, 1998).

CONCLUSION

This article presented a technique to create bi-dimensional maps from visual features detected over lines on a pair of images as a camera moves in a straight path with either constant speed or known positions. The necessary operations are very easy to carry out, and should be suitable for use in many real-time applications.

As a tool for obstacle detection, the technique presents the qualities of being able to identify suspended and negative obstacles. These are important elements in indoor environments but are rarely considered by the most simple obstacle detectors. The disadvantages of the technique are having to compare images taken from two different positions, and not being able to detect objects that are not directly in the path of the robot. But these limitations still fit the needs of many applications.

The theory presented here also offers a starting point for the development of more sophisticated mapping applications, and not just a local obstacle detection based on image pairs. The problem has many similarities to the classic one of fitting line segments to a set of non-associated points, and one possibility of research already being explored is the applications of classic techniques, such as the Hough, transform to perform mapping in the conditions described in this article (Werneck & Costa, 2010). Other possibilities being investigated is the independent estimation of camera orientation from the image gradient of only a few lines and columns of the image, and also edge extraction using this orientation

estimate and the same selected pixels (werneck & Costa, 2011).

REFERENCES

Barra, R., Ribeiro, C., & Costa, A. H. R. (2009). Fast vertical line correspondence between images for mobile robot localization. In *Proceedings of the 9th International IFAC Symposium on Robot Control* (pp. 153-158).

Borenstein, J., Everett, H. R., & Feng, L. (1996). *Navigating mobile robots: Systems and techniques*. Natick, MA: AK Peters.

Choi, S., Kim, T., & Yu, W. (2009). *Performance evaluation of RANSAC family.* Retrieved from http://www.bmva.org/bmvc/2009/Papers/Paper355/Abstract355.pdf

Civera, J., Davison, A., & Montiel, J. (2007). Dimensionless monocular SLAM. In J. Marti, J. Benedi, A. Mendonca, & J. Serrat (Eds.), *Proceedings of the Third Iberian Conference on Pattern Recognition and Image Analysis* (LNCS 4478, pp. 412-419).

Eade, E., & Drummond, T. (2009). Edge landmarks in monocular SLAM. *Image and Vision Computing*, *27*(5), 588–596. doi:10.1016/j.imavis.2008.04.012

Howard, A., & Kitchen, L. (1997). Fast visual mapping for mobile robot navigation. In *Proceedings of the IEEE Conference on Intelligent Processing Systems* (Vol. 2, pp. 1251-1255). Washington, DC: IEEE Computer Society.

Lenser, S., & Veloso, M. (2003). Visual sonar: Fast obstacle avoidance using monocular vision. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vol. 1, pp. 886-891). Washington, DC: IEEE Computer Society.

Murarka, A., Sridharan, M., & Kuipers, B. (2009). Detecting obstacles and drop-offs using stereo and motion cues for safe local motion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1646-1653). Washington, DC: IEEE Computer Society.

Nguyen, V., Gächter, S., Martinelli, A., Tomatis, N., & Siegwart, R. (2007). A comparison of line extraction algorithms using 2D range data for indoor mobile robotics. *Autonomous Robots*, *23*(2), 97–111. doi:10.1007/s10514-007-9034-y

Nourbakhsh, I. (1997). *Monologue on navigation.* Retrieved from http://goo.gl/016WH

Roosendaal, T., & Selleri, S. (2004). *The official blender 2.3 guide: Free 3D suite for modeling, animation and rendering.* San Francisco, CA: No Starch Press.

Solà, J. (2010, May 3-8). *Consistency of the monocular EKF-SLAM algorithm for three different landmark parametrizations.* Paper presented at the IEEE International Conference on Robotics and Automation, Anchorage, Alaska.

Strasdat, H., Montiel, J.M.M., & Davison, A.J. (2010). Real-time monocular SLAM: Why filter? *In Robotics and Automation* (ICRA), 2010 IEEE International Conference on, (pp.2657--2664). doi:10.1109/RO-BOT.2010.5509636

Sünderhauf, N., & Protzel, P. (2007). *Stereo odometry - a review of approaches* (Tech. Rep. No. 3/07). Chemnitz, Germany: Chemnitz University of Technology.

Trucco, E., & Verri, A. (1998). *Introductory techniques for 3-D computer vision.* Upper Saddle River, NJ: Prentice Hall.

Ulrich, I., & Nourbakhsh, I. R. (2000). Appearance-based obstacle detection with monocular color vision. In *Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence* (pp. 866-871).

Werneck, N., & Costa, A. H. R. (2010). *Monocular visual mapping with the Fast Hough Transform.* Paper presented at the Anais do 6th Workshop de Visão Computacional, Sao Paulo, Brazil.

Werneck, N., & Costa, A. H. R. (2011). Speeding up probabilistic inference of camera orientation by function approximation and grid masking. In *WSCG'2011 Communication Papers Proceedings* (pp. 127--134).