Workshop on Visual Localization for Mobile Platforms
(in association with CVPR)
Anchorage, Alaska, June 2008

Autonomous Navigation and Mapping Using Monocular Low-Resolution Grayscale Vision

Vidya N. Murali   Stanley T. Birchfield
Electrical and Computer Engineering Department
Clemson University,   Clemson, SC 29634
{vmurali, stb}@clemson.edu

Abstract

An algorithm is proposed to answer the challenges of autonomous corridor navigation and mapping by a mobile robot equipped with a single forward-facing camera. Using a combination of corridor ceiling lights, visual homing, and entropy, the robot is able to perform straight line navigation down the center of an unknown corridor. Turning at the end of a corridor is accomplished using Jeffrey divergence and time-to-collision, while deflection from dead ends and blank walls uses a scalar entropy measure of the entire image. When combined, these metrics allow the robot to navigate in both textured and untextured environments. The robot can autonomously explore an unknown indoor environment, recovering from difficult situations like corners, blank walls, and initial heading toward a wall. While exploring, the algorithm constructs a Voronoi-based topo-geometric map with nodes representing distinctive places like doors, water fountains, and other corridors. Because the algorithm is based entirely upon low-resolution (32×24) grayscale images, processing occurs at over 1000 frames per second.

1. Introduction

Psychological studies have shown that human intelligence does not require high-resolution images to ascertain information about the environment for basic navigation. The “selective degradation hypothesis”, developed by Owens and Leibowitz [2], says that some visual abilities such as vehicle steering and speed control remain relatively easy despite loss in visual acuity and color vision. For canonical tasks like walking or moving in a straight line, only a small percentage of what we see is actually useful, and in fact low-frequency information alone is sufficient for success. Motivated by this idea, we have developed a system that uses only low-resolution (32×24) grayscale images to navigate a previously unknown corridor environment and to produce a Voronoi-based topo-geometric map of the environment. By discarding 99% of the information captured from a 320×240 camera, the approach is computationally efficient, freeing up much of the CPU for other computation-intensive tasks, such as landmark recognition.

Navigating with a single camera is not easy. Perhaps this is why many approaches rely upon depth measurements from sonars, lasers, or stereo cameras to solve the problem. Granted, knowledge of distances to either wall, the shape of obstacles, and so on, would be directly useful for localizing the robot and building a geometric map of the environment. Lasers, however, are expensive and power-hungry, sonars cause interference, and stereo vision has its own difficulties (e.g., it requires texture to compute correspondence, is computationally expensive, and produces inaccurate results for many pixels). Indoor environments in particular often lack texture, rendering stereo matching an elusive problem in such places. In contrast, humans are quite good at navigating indoors with one eye closed, even with blurry vision, thus motivating us to find a different solution.

Our approach to navigation utilizes the ceiling lights and image entropy to keep the robot centered as it travels down the corridor. It is important to note that our approach does not require the lights to be of a certain shape or location – they may be in the center of the corridor, along both sides, or distributed uniformly across. When the lights are not visible, the robot determines whether it is at the end of the corridor using a combination of entropy, Jeffrey divergence, and time-to-collision. Turning is accomplished using the same measures. With these basic behaviors, the robot is able to autonomous wander an unknown indoor environment.

As the robot drives down the corridor, not all images captured are *salient*. Just as a human driving down a highway often experiences long stretches of monotonous scenery broken by intermittent landmarks, the robot perceives salient regions along either side of the corridor a small percentage of the time. In our approach to mapping, the measures of image saliency indicate the presence of a nearby landmark by a sudden increase in their value. These salient images correspond to locations of landmarks in the scene. The path followed by the robot along the corridor is automatically augmented with salient locations that become nodes in a Voronoi-based graph. Together, these form a map representing the topology of the environment which can be used for later localization and navigation tasks.

1.1. Previous Work

Vision-based mobile robot navigation has been studied by many researchers. From the early work of the Stanford Cart [29] to the current Aibo, navigation has been recognized as a fundamental capability that needs to be developed. According to the survey of DeSouza *et al*. [15], significant achievements have been made in indoor navigation, with FINALE [21] being one of the more successful systems. FINALE requires a model-based geometric representation of the environment and uses ultrasonic sensors for obstacle avoidance. NEURO-NAV [27] is another oft cited system that uses a topological representation of the environment and responds to human-like commands. The highly notable NAVLAB [36] and RHINO [7] are examples of proficient outdoor navigation systems which use a combination of vision and a wide variety of other sensors for navigation and obstacle avoidance. Moravec [29] and Nelson *et al*. [31], however, have emphasized the importance of low-level vision in mobile robot navigation, and Horswill [18] implemented a hierarchical and complete end-to-end vision-based navigational robot based on prior training of the environment.

One approach to navigation has been to use corridor lights, which can achieve robust navigation even in long corridors. In some systems, lights are used as landmarks in a teach/replay approach, with the camera pointing toward the ceiling [23]. The drawback of such a configuration, of course, is that the robot is blind to anything in front of it, not to mention that the system must be trained beforehand on the environment to be navigated. In another implementation, ceiling lights are used as aids in straight line navigation [22], but here again the camera points toward the ceiling, and the position and orientation of the rectangular lights are used for straight line navigation. Such a computation does not generalize well to environments in which the lights are not of a rectangular shape, or to robots with forward-facing cameras. Choi *et al*. [9] use a forward facing camera for detecting lights, but their approach also relies upon the actual geometrical features of the lamps, and it is restrained by the lights disappearing from the field of view, which is one of the main difficulties of forward-facing cameras. Our approach incorporates computations to handle this difficulty and to automatically detect and handle the end of a corridor, without any prior training of the environment or restriction on light shape.

With regard to mapping, the recent developments in Simultaneous Localization and Mapping (SLAM) have been based primarily upon the use of range sensors [28, 32, 4]. A few researchers have applied this work to the problem of building maps using monocular cameras, such as in the vSLAM approach [20], which is a software platform for visual mapping and localization using sparse visual features. An alternate approach is that of Davison *et al*. [14, 13], who also use sparse image features to build 3D geometric maps. In these visual SLAM techniques, either a complex matching process for a simple landmark representation [33] or a simple matching process for a complex landmark representation [34] is needed for robust robot localization. In indoor corridor environments, however, the lack of texture poses a major obstacle to such an approach. Indeed, popular techniques such as the Scale Invariant Feature Transform (SIFT) [34] or other feature representations have difficulty in such cases. Moreover, the computationally demanding nature of these algorithms often leaves little room for additional processing, and their design requires higher resolution images.

It is important to note that in most map-building systems, the robot is controlled manually. Autonomous mapping is rare, and autonomous vision-based mapping is even more rare [15]. Notable initiatives include the work done by Matsumoto *et al*. [24], who used omnidirectional cameras with stereo and optical flow to control navigation, and Shah *et al*. [35], who implemented an autonomous navigation system using a calibrated fish eye stereo lens system. However, these approaches require specialized cameras. Similarly, autonomous vision-based navigation is rare, with many techniques requiring a training phase in which the robot is controlled manually [3, 8, 25, 26, 19]. As a result, efficient autonomous map building of indoor environments using a single off-the-shelf camera has remained an elusive problem.

2. Autonomous Driving Down a Corridor

Our approach to autonomous driving in a corridor involves combining information from ceiling lights and entropy, as described in the following subsections.

2.1. Centering using corridor ceiling lights

The image is divided into four adjoining triangles defined by the two diagonals of the image. Assuming that ceiling lights reside in the top triangle, we use the mean horizontal location of the intensities above a threshold to determine whether the robot is traveling in the center of the corridor. By servoing on this location, the ability to navigate a long corridor with stability is achieved, even without any additional information from odometry or other sensors. This approach is not only simpler, but also more powerful and more general, than previous approaches that analyze the shape of lights. For example, Figure 1 shows a variety

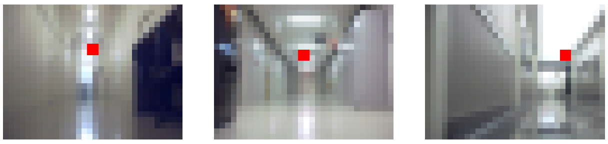

Figure 1. Different ceiling lights and their mean locations (red dots) detected by our algorithm. Notice that there is no restriction on the shape or location of lights; in the right image the lights are on the sides of the corridor pointing toward the reflective ceiling.

of lights that are successfully detected using this method. Note that ceiling lights provide an added advantage over vanishing points because they are affected by translation, thus enabling the robot to remain in the center of the corridor while also aligning its orientation with the walls.

2.2. Distinguishing the corridor by scalar entropy

The entropy of an image is a scalar representing the statistical measure of randomness that can be used to characterize its texture:

$$H(K) = \sum_{p\in K} -p\log p, \tag{1}$$

where p is the count value for each bin in the histogram K of the image I (256 bins for a graylevel image). It is also a measure of the information content in an image. When the robot approaches a planar surface, like a blank wall or the surface of an untextured or structured object, the entropy drops; this implies that the camera is facing a planar obstacle immediately in front of it. In richly textured images, time-to-collision (TTC) [1] or central image divergence threshold [11] can be used to determine the position of a frontal obstacle. But in an environment devoid of texture and consisting of uniform color, these methods will fail. Using entropy (in addition to the existing methods), therefore, is a promising way to react to a situation where the image does not provide enough information for navigation. Other researchers have used entropy for determining the direction of navigation and for global visual localization using omnidirectional images [6, 16].

Entropy is used in several ways. While driving, the entropy values on the two sides of the image are compared, and if either of the values drops sharply, the robot turns away from the side with the lower entropy. In the same manner, while turning at the end of a corridor, the robot continues turning as long as either side has low entropy and the overall entropy is below a threshold. That a low entropy value indicates a nearby wall is illustrated in Figure 2, where sharp drops in entropy correspond to images where the robot is facing a blank wall.

Entropy can also be used to find corridors. Figure 3 shows a plot of entropy values as the robot turns on the spot facing three branches of a T-junction. The entropy is high when the robot is aligned with the corridor, and it drops sharply when the robot faces the wall. Therefore, entropy can be used to detect the presence of an open corridor for navigation when other metrics fail, whether in textured or untextured environments.

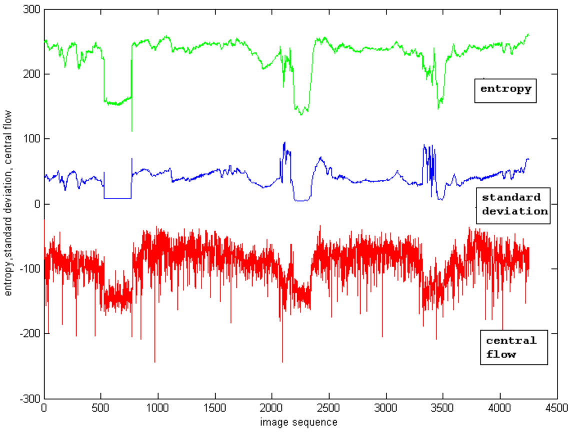

Figure 2. Comparison of image entropy, absolute image standard deviation, and central optical flow of the image, all measured while the robot traveled in a building. The three drops in entropy correspond to three turns, when the robot faced the walls. Notice that the entropy values are more easily distinguished (and less noisy) than those of the other measures.

2.3. Homing mode

When the robot nears the end of a corridor, the lights disappear from the camera's field of view and the overall entropy drops. When either of these occurs, the robot automatically captures the current image and stores it as the 'home' image. Keeping that image in view, the robot navigates toward it using *homing* [30]: The current image is compared with the home image after shifting left and right by a maximum disparity of one pixel. The result that yields the lowest sum of absolute difference (SAD) indicates the robot's direction of motion. This keeps the robot in the center of the corridor even when the lights are not visible.

3. Detecting the End of the Corridor

The end of the corridor is determined by combining three measures: entropy (just described), relative entropy, and the time-to-collision, in order to navigate in different indoor environments with different levels of texture/information and lighting.

3.1. Relative entropy

If we consider two discrete distributions with probability functions p_k and q_k, then the Kullback-Leibler distance of

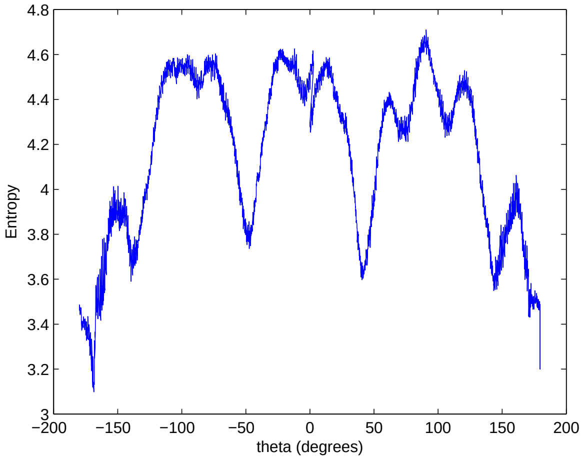

Figure 3. Entropy captured as the robot turned in place at the T-junction of two corridors. Entropy is high when the robot faces the length of a corridor and drops sharply on either side, so the three peaks indicate the three corridor directions. Maintaining high entropy allows the robot to avoid the specular reflections of the walls.

p with respect to q is given by

$$D\left(p,q\right)=\sum_{k}p_k\log\left(\frac{p_k}{q_k}\right), \qquad (2)$$

which is a measure of the distance between two distributions. In our application, p_k and q_k represent the intensity histograms of two images, so that the relative entropy D measures how different one image is from the other. One drawback of the Kullback-Leibler measure is that it is not a true distance, because $D(p,q)\neq D(q,p)$. For a symmetric measure, the Jeffrey divergence is used:

$$J\left(p,q\right)=\sum_{k}\left(p_k\log\left(\frac{p_k}{q_k}\right)+q_k\log\left(\frac{q_k}{p_k}\right)\right). \qquad (3)$$

Jeffrey divergence has been used previously for vision based robot localization for comparing color histograms in typical SLAM algorithms and has been shown to be a good metric for histogram comparison [37].

As the robot moves toward the end of the corridor in the homing phase described in the previous section, every image is compared with the home image using Jeffrey divergence. This measures the amount of relative information between the two images, i.e., how different one image is from the other. The divergence value increases steadily while the home image is in view, then the value increases rapidly as the robot approaches the end of the corridor. This rapid change signifies that the current image is no longer recognizable as ‘home’ (see Figures 4 and 5).

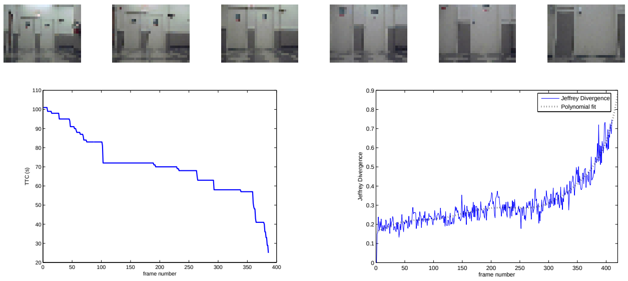

Figure 4. Time-to-collision and Jeffrey divergence for an image sequence in which the robot approaches a pair of doors. TOP: Sample images from the sequence. BOTTOM: Plot of the TTC (left) and Jeffrey divergence (right) versus time. The former decreases, while the latter increases; combining the two enables robust detection of the end of a corridor.

3.2. Time-to-collision detector

Time-to-collision (TTC) is defined as the time taken by the center of projection of a camera to reach the surface being viewed, if the relative velocity remains constant [17]. Horn *et al*. [17] have recently described a novel method to determine the time-to-collision using image brightness derivatives (temporal and spatial) without any calibration or tracking. This method computes the time to contact with just two frames of a sequence. Although each individual estimate is noisy, a filtered version of the output yields a reliable estimate as the camera approaches the object. Of specific importance is the case of a planar surface for which the algorithm is a simple one and can be applied to the case of a robot approaching the end of a corridor. For the case of translation motion along the optical axis towards a plane perpendicular to the optical axis, the TTC is given by

$$TTC=\frac{-\sum(G\left(x,y\right))^2}{\sum G\left(x,y\right)E_t}, \qquad (4)$$

where $G\left(x,y\right)=xE_x+yE_y$, E_x and E_y are spatial image brightness derivatives, E_t is the temporal derivative, and the sum is over the desired planar object (in some cases the entire image) [17]. Figures 4 and 5 show that the TTC increases as the robot approaches the end of a corridor. By combining Jeffrey divergence and TTC, the end of a corridor can be detected reliably.

3.3. Turning at the end of a corridor

The robot displays tropism at the end of each corridor, making an autonomous decision to turn in order to find the new adjacent corridor. While turning, the robot searches for ceiling lights and high overall entropy. The robot enters a rotational search mode until it finds another source light in the ceiling. If it sees the light, it corrects its course and follows the light into the new corridor following the same procedure as above. However, if it does not see any lights

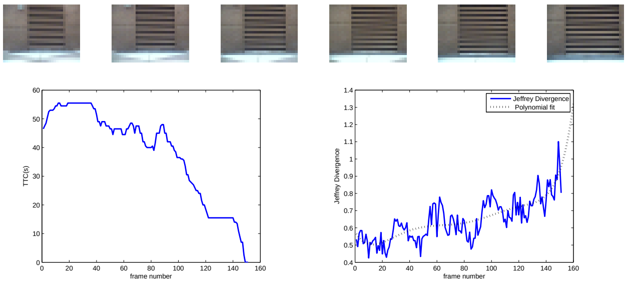

Figure 5. Time-to-collision and Jeffrey divergence for an image sequence in a textured environment in which the robot approaches a brick wall with a ventilator. TOP: Sample images from the sequence. BOTTOM: Plot of the TTC (left) and Jeffrey divergence (right) versus time. It can be seen that the metrics describe the approaching end successfully in both textured ('information rich') and relatively textureless environments.

on all sides but still senses the presence of a corridor indicated by an entropy value greater than a threshold (see Figure 3), then it navigates in that direction using 'homing' as described above and the process continues. If lights come into view again, the robot follows the light.

4. Autonomous Mapping

The same metrics that were used for navigation can be used to determine *distinctive/salient* landmarks for map building in an incremental process. Boada *et al*. [5] have implemented a popular framework for Voronoi-based maps and localization. The Voronoi-based maps are roadmap methods and are preferred for corridor mapping because of their *accessibility, connectivity, and departability* [10] and can be constructed incrementally by the robot. In this approach, the graph consists of the links which represent the obstacle-free path followed by the robot and the nodes which represent the *distinctive/salient* places along the path.

4.1. Joint Probability Distribution of Distinct Landmark Measures

For landmark detection only one-sixth of the image is considered on either side, because this narrow region contains landmarks as seen along a corridor. This further implies that only 33% of the 32×24 image is used. We determine distinct landmarks along the hallway by using the measures of image scalar entropy and relative entropy between two subsequent images. Let X represent a normalized random variable representing the entropy of the gradient magnitude of i^{th} image seen along the hallway, and let Y represent the Jeffrey divergence between the i^{th} and the $(i-1)^{th}$ image gradients. Then the Joint Probability Density (JPD) of the two variables represents the *distinctiveness* measure of the image:

$$P_{xy}\left(X,Y\right) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\left(\frac{X^2}{2\sigma_x^2}+\frac{Y^2}{2\sigma_y^2}\right)\right]. \quad (5)$$

This can be described as a measure of how information-rich and unique an image is. A landmark is therefore defined as an image that has *interesting*, *recognizable* information that is distinct from the previous image. It is assumed that two consecutive frames in the sequence do not have two different potential landmarks. Considering the speed of the robot and the capture rate of the camera, this is highly unlikely.

Because the relative entropy between two images is independent of the absolute entropy of either one, X and Y can be considered as independent variables. Regional maxima on the JPD give locations/images that represent landmarks. It can be seen from the results in Figure 9 that even in images of low resolution (where traditional point features are hard to detect/track) the simple measures indicated above give a clear indication of a landmark. The algorithm does not represent each landmark uniquely (which would be difficult in a typical indoor environment consisting of corridors with identical doors) but instead represents locally the presence of a landmark.

5. Experimental Results

The algorithm was tested on an ActivMedia Pioneer P3AT mobile robot platform equipped with a forward-facing Logitech Quickcam Pro4000 webcam in three floors of a building on our campus. For historical reasons, the three floors do not share the same appearance in terms of the color of the walls, the placement of the lights, the locations of the doors, the locations of nearby corridors, and so on. In particular, the corridors have entirely different lighting conditions, ranging from a single row of fluorescent lamps to sodium vapor lamps to lights on either sides of the corridor ceiling (see Figure 1). The texture (information content) in the corridors is also different, with the basement having textureless walls and floors of uniform color (see Figure 6). Only the grayscale information from the 32×24 downsampled images from the camera was used.

On all three floors the robot autonomously navigated the corridors, turning at the end of each corridor using the algorithm described. At the end of a corridor, the direction of turning was chosen at random if both options were available; otherwise the robot turned in the open direction. Figures 7, 8, and 9 show the path taken by the robot on two of the floors, overlaid on a hand-constructed map of the environment to provide context for interpreting the results. On the first floor the robot turned left twice at the end of each corridor; in the basement the robot turned right, then turned left (arbitrarily), navigated to the end of the corridor, then turned around 180 degrees and headed back down the last

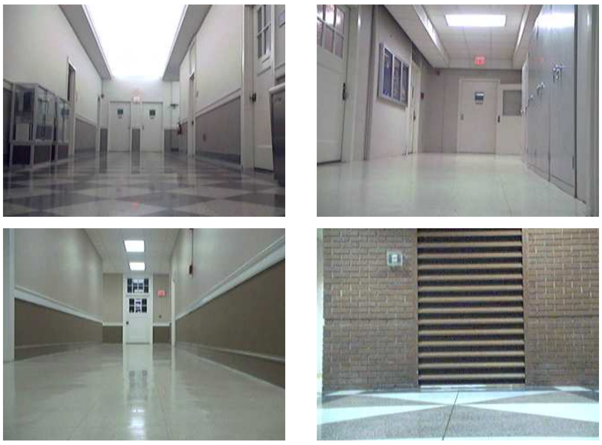

Figure 6. Example experimental sites shown in high resolution to reveal the difference in texture and lighting.

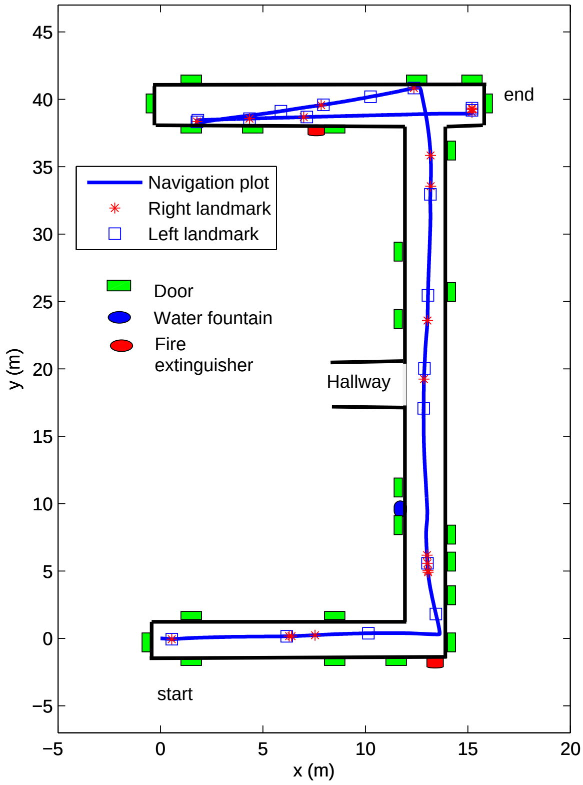

Figure 7. Automatically computed Voronoi-based map with nodes representing the approximate distinctive landmarks on the third floor of the building. It can be seen that the landmarks have been verified by the returning robot in the top wing of the corridor.

corridor in the opposite direction. In all cases the robot remained in the center of the corridor, avoiding collision with the walls or obstacles.

Since the robot's odometry is prone to drift over large distances, these plots include an effective method to reduce the drift using the motor commands given by the vision module. Inspired by the work of Crowley [12], which combines the measured position and the expected position based on motor commands using a Kalman filter and a retroactive odometric correction using sensor fusion [23], we use the motor commands issued by the vision module to incrementally correct the odometry. Though drift errors persist to a small degree, this incremental method is sufficient for the general purpose of this initiative.

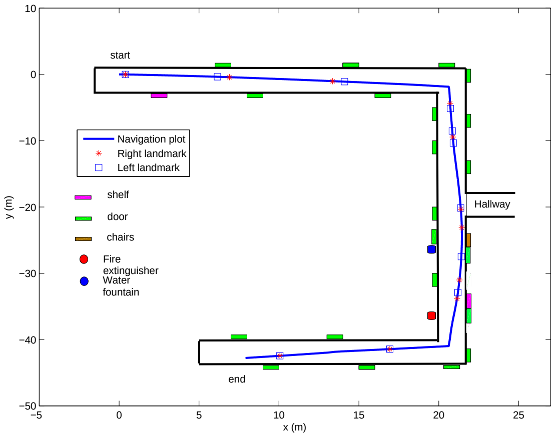

Figure 8. Automatically generated Voronoi map of the first floor of the building.

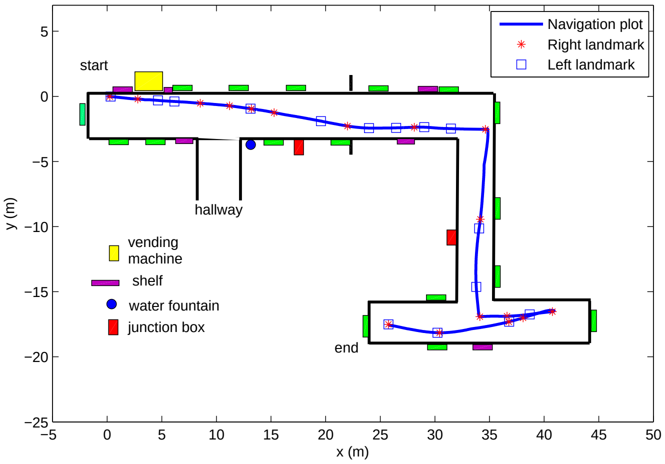

Figure 9. Automatically generated Voronoi map of the basement of the building.

Figures 7, 8, and 9 also show the generated Voronoi-based map overlaid. Most of the important landmarks have been captured. The nodes represented in the Figure 7 represent distinctive regions along the corridor of the third floor. This map is a Voronoi-based topological representation built autonomously. With odometry combined it can be described as a topo-geometric map similar to the description in [5] as it combines real distances with the skeleton. The

Location	N_L	N_D	F	M
Floor 1	12,11	10,10	2,0	2,1
Floor 3	12,13	14,13	4,3	2,3
Basement	15,14	15,13	1,2	1,1

Table 1. Quantitative landmark detection results. From left to right: the number of landmarks N_L, the total number detected by the algorithm N_D, the number of false landmarks detected F, and the number of landmarks missed by the algorithm M. Each cell in the table contains the number for left and right, separated by a comma.

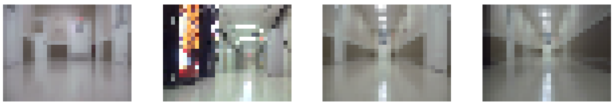

Figure 10. Landmark images containing a landmark on the left side of the image.

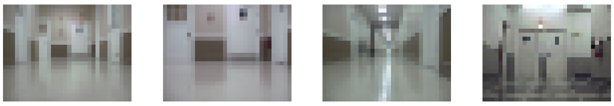

Figure 11. Landmark images containing a landmark on the right side of the image.

landmarks seen to the left of the robot are represented by a square, and the landmarks seen on the right are represented by an asterisk. At corridor junctions it can be seen that left and right landmarks overlap. This is because the robot turns at junctions to search for lights. Furthermore, the multiple doors at junctions are recognized as one landmark because they are all captured during the rotation of the robot at junctions. It is interesting to observe the top wing of the corridor in Figure 7. The left and right landmarks validate each other because the robot returns along the same path in the opposite direction. Some example images representing the detected landmark positions are shown in Figures 10 and 11.

Table 1 shows the analysis of the results. The landmarks are counted in the order of the robot's navigation path, while the returning landmarks are not counted. Also note that in some cases two entities that are immediately next to each other are detected as one distinct region/landmark (e.g., a door with an adjoining shelf on the wall). It can be seen that in the worst case at least 70-80 % of the landmarks are detected successfully.

The algorithm is efficient, capable of running at over 1000 frames per second. Therefore with a standard 30 Hz camera, the algorithm consumes approximately 3% of the CPU, thus freeing the processor for other concurrent tasks. In our experiments, the robot was run indoors at a moderate speed of 0.4 m/s.

6. Conclusion and Future Work

The navigational behavior of a mobile robot is modeled by a set of paradigms that work in conjunction to correct its path in an indoor environment based on different metrics. Special emphasis is placed on using low resolution images for computational efficiency and metrics that capture information content and variety that cannot be represented using traditional point features and methods. The resultant algorithm enables end-to-end navigation in indoor environments with self-directed decision making at corridor ends, without the use of any prior information or map. The system forms the basis of an autonomous mapping system that is built using the same low resolution metrics to present a Voronoi-based topo-geometric map that can be used for robot localization.

Future work involves the development of a layered approach where higher resolution image processing will augment the system to handle complex requirements like landmark matching. The Joint Probability Distribution can be made more robust using multiple temporal derivatives and smoothing. This autonomous mapping can seamlessly integrate with existing topological localization modules that use Jeffrey divergence to match landmarks.

References

[1] N. Ancona and T. Poggio. Optical flow from 1D correlation: Application to a simple time-to-crash detector. In *Proceedings of the 4th International Conference on Computer Vision*, 1993.

[2] J. Andre, D. A. Owens, and L. O. Harvey, Jr., editors. *Visual perception : The influence of H. W. Leibowitz*. Washington, DC: American Psychological Association, 2003.

[3] G. Blanc, Y. Mezouar, and P. Martinet. Indoor navigation of a wheeled mobile robot along visual routes. In *Proceedings of the International Conference on Robotics and Automation*, pages 3354–3359, 2005.

[4] J.-L. Blanco, J.-A. Fernández-Madrigal, and J. Gonzalez. A new approach for large-scale localization and mapping: Hybrid metric-topological SLAM. In *Proceedings of the International Conference on Robotics and Automation*, 2007.

[5] B. L. Boada, D. Blanco, and L. Moreno. Symbolic place recognition in Voronoi-based maps by using hidden Markov models. *Journal of Intelligent and Robotic Systems*, 39(2):173–197, 2004.

[6] B. Bonev, M. Cazorla, and F. Escolano. Robot navigation behaviors based on omnidirectional vision and information theory. *Journal of Physical Agents*, 1(1):27–35, September 2007.

[7] J. M. Buhmann, W. Burgard, A. B. Cremers, D. Fox, T. Hofmann, F. E. Schneider, J. Strikos, and S. Thrun. The mobile robot RHINO. *AI Magazine*, 16(2):31–38, 1995.

[8] Z. Chen and S. T. Birchfield. Qualitative vision-based mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2006.

[9] K. Choi, S. Bae, Y. Lee, and C. Park. A lateral position and orientation estimating algorithm for the navigation of the vision-based wheeled mobile robot in a corridor. In *SICE 2003 Annual Conference*, volume 3, 2003.

[10] H. Choset, I. Konukseven, and A. Rizzi. Sensor based planning: A control law for generating the generalized Voronoi graph. In *Proceedings of the IEEE International Conference on Advanced Robotics*, 1997.

[11] D. Coombs, M. Herman, T. Hong, and M. Nashman. Real-time obstacle avoidance using central flow divergence and peripheral flow. In *Proceedings of the 5th International Conference on Computer Vision*, July 1995.

[12] J. L. Crowley. Asynchronous control of orientation and displacement in a robot vehicle. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 1277–1282, 1989.

[13] A. Davison and N. Kita. Sequential localization and map-building for real-time computer vision and robotics. *Robotics and Autonomous Systems*, 36(4):171–183, 2001.

[14] A. Davison and D. Murray. Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):865–880, July 2002.

[15] G. N. DeSouza and A. C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, 2002.

[16] F. Escolano, B. Bonev, P. Suau, W. Aguilar, Y. Frauel, J. Saez, and M. Cazorla. Contextual visual localization: cascaded submap classification, optimized saliency detection, and fast view matching. In *IEEE International Conference on Intelligent Robots and Systems*, 2007.

[17] B. K. Horn, Y. Fang, and I. Masaki. Time to contact relative to a planar surface. *IEEE Intelligent Vehicles Symposium*, pages 68–74, June 2007.

[18] I. Horswill. Specialization of perceptual processes. Technical Report AITR-1511, MIT-AI, 1994.

[19] S. D. Jones, C. S. Andersen, and J. L. Crowley. Appearance based processes for visual navigation. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 551–557, 1997.

[20] N. Karlsson, E. D. Bernard, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich. The vSLAM algorithm for robust localization and mapping. In *Proceedings of the International Conference on Robotics and Automation*, pages 24–29, 2005.

[21] A. Kosaka and A. C. Kak. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. *CVGIP: Image Understanding*, 56(3):271–329, 1992.

[22] F. Launay, A. Ohya, and S. Yuta. Image processing for visual navigation of mobile robot using fluorescent tubes. In *Proceedings of the International Conference on Circuits/Systems, Computers and Communications*, volume C5-1, pages 664–667, July 2001.

[23] F. Launay, A. Ohya, and S. Yuta. A corridors lights based navigation system including path definition using a topologically corrected map for indoor mobile robots. In *Proceedings of the International Conference on Robotics and Automation*, volume 4, pages 3918–3923, 2002.

[24] Y. Matsumoto, K. Ikeda, M. Inaba, and H. Inoue. Exploration and navigation in corridor environment based on omni-view sequence. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 2, pages 1505–1510, 2000.

[25] Y. Matsumoto, M. Inaba, and H. Inoue. Visual navigation using view-sequenced route representation. In *Proceedings of the International Conference on Robotics and Automation*, volume 1, pages 83–88, 1996.

[26] Y. Matsumoto, K. Sakai, M. Inaba, and H. Inoue. View-based approach to robot navigation. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 545–550, 2000.

[27] M. Meng and A. Kak. NEURO-NAV: a neural network based architecture for vision-guided mobile robot navigation using non-metrical models of the environment. In *Proceedings of the IEEE International Conference on Robotics and Automation,*, volume 2, pages 750–757, 1993.

[28] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fast-SLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002.

[29] H. Moravec. Locomotion, vision and intelligence. In M. Brady and R. Paul, editors, *Robotics Research: The First International Symposium*, pages 215–244. Cambridge, Massachusetts: The MIT Press, Aug. 1984.

[30] R. C. Nelson. *Visual navigation*. PhD thesis, University of Maryland at College Park, College Park, MD, USA, 1988. Director-Yiannis Aloimonos.

[31] R. C. Nelson and J. Aloimonos. Using flow field divergence for obstacle avoidance towards qualitative vision. In *Proceedings of the 2nd International Conference on Computer Vision*, pages 188–196, 1988.

[32] F. Ramos, J. Nieto, and H. Durrant-Whyte. Recognising and modelling landmarks to close loops in outdoor SLAM. In *In Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, pages 2036–2041, Apr. 2007.

[33] A. Ranganathan, E. Menegatti, and F. Dellaert. Bayesian inference in the space of topological maps. *IEEE Transactions on Robotics*, pages 92–107, 2006.

[34] S. Se, D. Lowe, and J. J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, pages 364–375, 2005.

[35] S. Shah and J. K. Aggarwal. Mobile robot navigation and scene modeling using stereo fish-eye lens system. *Machine Vision and Applications*, 10(4):159–173, 1997.

[36] C. Thorpe. *Vision and Navigation, the Carnegie Mellon NAVLAB.* Kluwer, 1990.

[37] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings of the International Conference on Robotics and Automation*, pages 1023–1029, Apr. 2000.