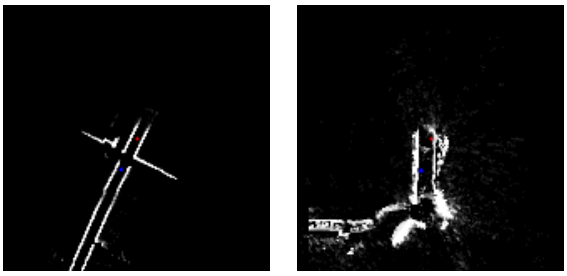
CONFIDENTIAL. Limited circulation. For review only.

Outdoor Intersection Detection for Autonomous Exploration

Piyoosh Mukhija[1] Siddharth Tourani[2] K Madhava Krishna[3]

***Abstract*— In the field of outdoor navigation, there still remain largely unsolved problems. In this paper we address the problem of detecting road intersections. We present two approaches to solve the problem of intersection detection in an unstructured outdoor setting. The first is a natural extension of the popular VFH* obstacle avoidance algorithm. It detects intersections and tracks, over a period of time, the angles at which gaps in the robot's certainty grid (CG) are first observed. The second approach uses techniques from image processing and computational geometry on the certainty grid image, to extract a skeleton of the navigable region, thus providing the intersections. We show experimental results portraying intersection detection due to both methods and show the results. On the whole, we found that the robot was able to detect all possible intersections.**



(a) (b)

Fig. 1. A comparison of indoor and outdoor certainty grid images. (a) The indoor certainty grid has sharply defined edges and contours. (b) The outdoor certainty grid in comparison has much more noise.

I. Introduction

Most of the literature on autonomous exploration has concentrated on the indoor setting. Exploring an unstructured outdoor environment is a far more challenging problem for a robot. Solving them can lead to the development of various outdoor robotic applications such as search and rescue in bio-hazard regions, surveying and security and planetary exploration.

Unstructured outdoor environments pose various challenges for an autonomous vehicle. When compared with the indoor environments, distances are greater, color and lighting conditions are more varied, the ground surface is never level, and in a populated region there are moving obstacles. For autonomous outdoor exploration, a robot must be able to handle all of the above.

In order for any outdoor exploration to be complete, the exploring robot must be able to detect the intersections to the road it is traversing. Without intersection detection, the exploration is left incomplete, i.e there will remain unexplored regions in the environment.

Intersection detection is a hard problem. One cannot rely on visual cues like sign boards because they are not present at every intersection. Indoors the surface on which the robot moves is usually flat and smooth. Any data obtained from sensors in such a setting will give reasonably accurate information about the environment. In contrast outdoors, uneven road surfaces can lead to noisy sensor readings, leading to an inaccurate representation of the environment.

In our system, the sensor data is fused into a certainty grid image. For an indoor environment, the certainty grid image is characterized by precise contours with sharp and straight edges. Thus, for intersection detection in an indoor setting, one can simply find gaps between edges and mark them. Fig. 1(a). shows how such a simple approach for edge detection would never work for the noisy certainty grid image of an outdoor environment. Figures 1(a) and 1(b) illustrate the situation corresponding to an intersection detection in indoor and outdoor settings. While indoor laser maps provide for crisp obstacle locations and gaps, outdoor maps show a much larger uncertainty. This uncertainty stems from, amongst other things, due to (i) calibration errors that determine the laser pose with respect to camera, (ii) with a tilted laser pointing to the ground it always returns a reading unlike indoors where free space is easily detected by laser returning a maximum reading, (iii) sub-optimal means of weighing camera and laser data when fusing onto a certainty grid, (iv) the dimensionality increase in representation of vehicle pose. Due to this the entropy of the regions around the intersection and the overall certainty grid is considerably more than in an indoor setting. Robust state estimation within a SLAM framework could be one way of alleviating some of the issues. Since the problem is one of intersection detection for road network exploration, wherein a topological map of roads and intersections would suffice, SLAM is not a concern. We propose two novel methods for intersection detection under the duress of suhc large uncertainty regions in the map.

Due to the reasons mentioned above, there may be high certainty values concentrated in a small number of cells for some obstacles, while low certainty values may be spread over a large number of cells for others. Thus, instead of considering individual pixel values, we are limited to a statistical method operating over a region of pixels. As the component data for certainty grid input is mainly range data from the LRF and and range inferred from the single camera image , the desired statistical method should account for uncertainties in obstacle ranges from the robot positions. As

All authors are with Robotics Research Centre IIIT Hyderabad, India. piyoosh@mukhija.info, siddharth.tourani@research.iiit.ac.in, mkrishna@iiit.ac.in

Preprint submitted to 15th International IEEE Annual Conference on Intelligent Transportation Systems. Received March 15, 2012.

with the VFH*[1] algorithm we make use of polar certainty histograms. The polar histograms in these techniques are generated by an integration of certainty values along the rays from the robot and another integration of a set of rays into a sector, thereby identifying free directions around the robot for navigation, without requiring an exact range estimation to obstacles.

Hence, these techniques are further adapted/extended as our first algorithm to identify intersections on a road being travelled. First, in the noisy environment, the algorithm tries to identify the gaps in the road sides using the polar histogram. Next, the algorithm tracks these gaps in order to accumulate evidence of the presence of an intersection. If there is sufficient evidence for an intersection it is stored and marked. If due to noise, a non-existent gap is detected, it is unlikely that the same gap will de detected in subsequent scans, and so will not be marked as an intersection. Since two of the detected gaps correspond to the forward and backward direction of the road, detection of the same set of three or more gaps over multiple such views indicates the presence of an intersection. The directions of these gaps represent the directions of the road branches from the intersection point as well.

Our second method, uses a multi-resolution wavelet analysis followed by a modified Hough transform and skeletonization. As input the algorithm receives a pictorial representation of the environment, in the form of a certainty grid image. A multi-resolution analysis checks for the invariant features in the image at different scales. This helps to eliminate the noise or blurred influence in the image and detect the edges corresponding to the borders of the road. A modified hough transform is then used to detect the borders of the road in the image. The last step is skeletonization, after which the we get the intersections, by finding the points at the intersections.

A. Related Work

Most autonomous exploration systems have used frontiers as the basis for their exploration. They have been used in various forms. The seminal work of Yamauchi used a frontier-based exploration[2]. Most of the exploration algorithms that came after this were also frontier-based.

They have varied from using greedy strategies to explore the frontiers [3], to more sophisticated heuristics or semantic cues to guide the exploration [4], [5]. Most recently, a PDE based exploration that enforces boundary conditions using the frontiers co-ordinates was proposed in [6].

For exploring a network of roads detection of intersections where the road forks become more critical than frontier detection. Frontiers prove to be sufficiently low level primitive outdoors, whereas one is primarily interested in a higher level primitive/ topological construct such as an intersection.

After an extensive literature search, we were unable to find many papers that addressed the problem of intersection detection explicitly. The literature we found most relevant to our problem came from geo-spatial images, where techniques were developed to detect road networks from images.

These techniques can roughly be classified into 3 types:

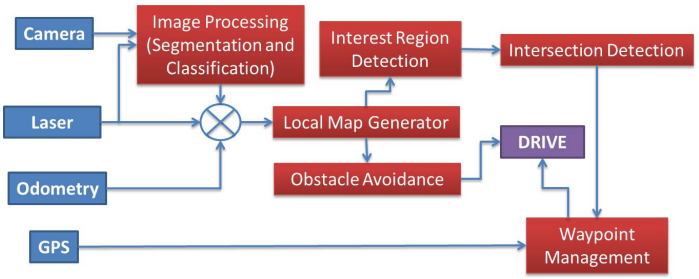


Fig. 2. The modules that make up the system and their various interconnections.

- Edge-Based
- Morphological
- Gradient-Based

Edge-based methods are the most commonly used. They are simple and computationally unintensive but may contain spurious detections due to their high sensitivity to noise. They work best on images containing orthogonal intersections, but fail at roundabouts.

Morphological methods, as used in [7], [8], use morphological operators such as the dilation and erosion operators to identify the roads in the image. These methods are also computationally unintensive. These methods unfortunately yield incomplete detection of the road networks and spurious detections. Other methods involving gradients of the pixels also have similar problems.

Another line of research involves matching a reference profile to a region in the image [10] or getting the robot to learn the various possible intersections [9]. Such methods, yield better results than the others, but fail when the image obtained is itself noisy. Also, they are usually computationally intensive and require a large training set.

Our algorithms, in comparison require no training data and make no assumptions as to the shape of the road.

B. System Overview

The system used consists of several modules working asynchronously. Fig. 2 displays the various modules and the interconnections between them. The camera looks in front of the vehicle and is tilted such that the image contains a large part of the road ahead as well as obstacles. The frames captured from the camera are sent to the image processing module to differentiate between navigable and non-navigable terrains. The Image Processing Module takes simultaneous inputs from both Camera and Laser Range-Finder (LRF). The LRF is also tilted (approx 4.2 degrees) in order to provide a 3D-style obstacle and road point cloud (upto 5 metres ahead of robot) by the virtue of robot's forward motion (see Fig. 3). The tilted LRF and a subset of resulting point cloud determine free ground region(s) in front of the camera, which are projected onto camera image through known calibration and orientations of LRF and camera Fig. 4.a. These projected regions serve as training samples for Gaussian Mixture Models algorithm. These Gaussians form the possible template road colors on the basis of which

Preprint submitted to 15th International IEEE Annual Conference on Intelligent Transportation Systems. Received March 15, 2012.

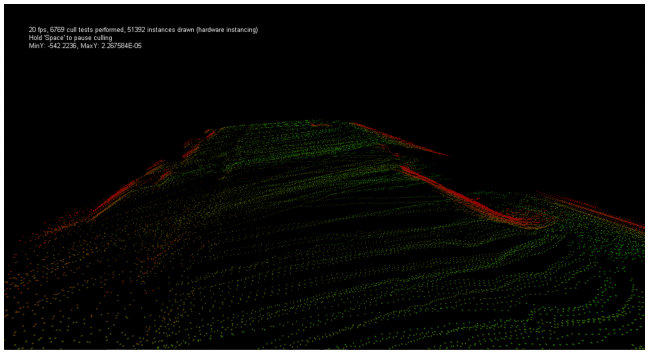


Fig. 3. A 3D Point Cloud of the Laser

the image is segmented into navigable and non-navigable terrains. Such a method allows for adaptation to the different colors and textures a road might possess very quickly. A similar method was used in [11].

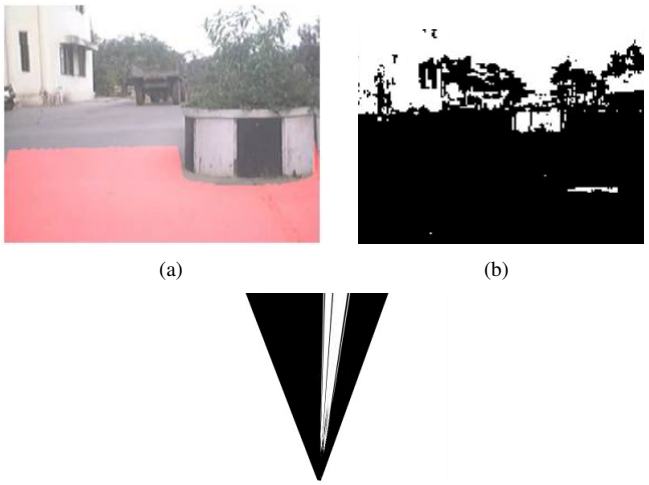
The segmented image along with the laser data and robot encoder readings are used to generate a certainty grid of the robot's local environment. Messages from the GPS are stored in the waypoint management module. The Obstacle Avoidance Module generates steer messages using the GPS data. It consists mainly of an implementation of the VFH* algorithm. The stored waypoints are later used by the Waypoint Management Module to play a strategy for exploration.

The road mapping problem does not require rigorous estimation of the robot's state as well as it's surrounding environment in terms of precise metric co-ordinates, quite unlike an outdoor SLAM system. The locations of the robot when it detects intersections are stored by GPS readings with an accuracy of 2m. While revisiting these positions the robot is able to identify these intersections again and move onto another branch of the road.

C. Data Fusion

This section contains how the data from the LRF and camera are fused. The calibrations of the laser and camera are known to us, as are their relative orientations. We use laser scan data to determine the free space regions in front of the robot. These regions are then projected onto the image. To detect the road a subset of this region is selected as our sampling region on the basis of which we train our Gaussian Mixture Models algorithm. These Gaussians form the possible template road colors on the basis of which segmentation is done. Such a method allows for adaptation to the different colors and textures a road might possess quickly. A similar method was used in [11].

We use data from two sensors to generate the certainty grid for the following reasons. The camera is used to detect the obstacles from distances greater than LRF (providing an early population of certainty grid) and the ground level non-traversible regions on the sides of road (like mud, grass, etc). However, the camera based detection is noisy (has range errors) and frequently produces false positives or negatives. Also, it needs to learn the actual current appearance of road which varies greatly when travelling outdoors. Thus the



(a) (b) (c)

Fig. 4. (a): The region known to be ground from laser is marked here with a translucent overlay. (b): Segmentation and Classification result while approaching a divider. The black parts of the image represent road. (c): Top-View of the Segmented Image generated by homography.

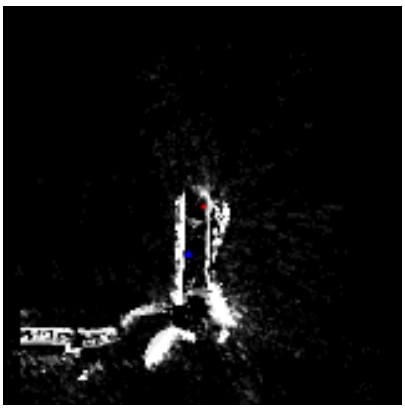


Fig. 5. The certainty grid. The red dot is the current position of the robot. The blue dot is the position that the gap-tracking algorithm receives.

LRF is able to identify obstacles at shorter distances and is able to determine their positions and more accurately than camera. Also, it is able to provide a training dataset for the image processing module and allows it to quickly adapt to the changing road appearance. Thus, the regions wrongly classified by one sensor are correctly classified by the other. A laser scan of a non-navigable region like mud would not be detected as obstacle, but, the camera on the basis of a marked color difference would classify it as an obstacle. Likewise, a grayish obstacle on the road would be classified as road by the camera but be detected as an obstacle by the laser. The certainty grid image from the fused data serves as input for the intersection detection module. Fig. 5 shows a certainty grid image.

II. Intersection Detection Based On Gap Angle Tracking

Through accumulated experience over several outdoor experiments we found that the intersection detection is best

Preprint submitted to 15th International IEEE Annual Conference on Intelligent Transportation Systems. Received March 15, 2012.

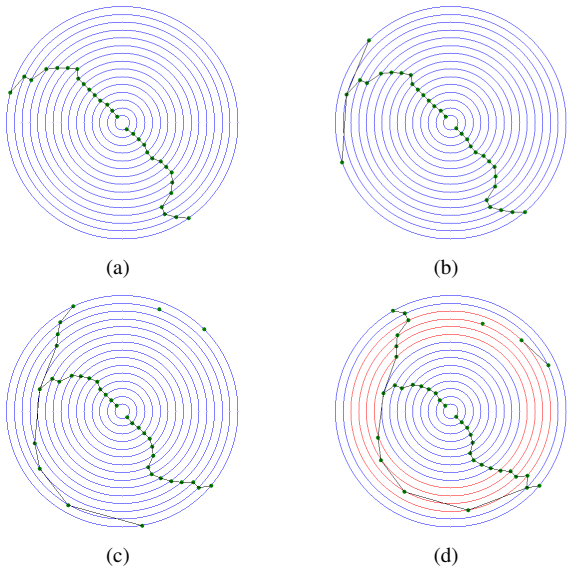


Fig. 6. The process of gap tracking and intersection detection is shown in these figures. Top-Left: Initially there are only two gaps being tracked. The one in the front of the robot, and, the one behind it. Top-Right: The first sighting of a gap on the left is shown. The path ahead of the robot bifurcates. Bottom-Left: The path continues being tracked. Bottom-Right: When the path has been tracked for long enough, the intersection is detected. The red circles Show that the intersection has been detected.

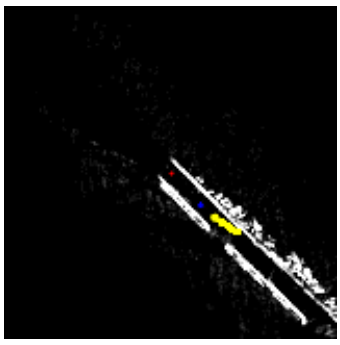


Fig. 7. The corresponding intersection detected by the gap-tracking algorithm. The intersection is marked by yellow. Note the similarity in alignment between the chains in the polar histograms and the certainty grid image.

when it is delayed, in other words it is easier to detect the intersection once the robot has already passed through the intersection. For this purpose we consider locations visited by the robot (points from robot's trajectory) some time instances into the past as candidate locations from where to detect gaps. We denote the extracted portion of certainty grid within a given radius of the robot's chosen past location (P) as critical region (CR).

Our algorithm can be thought of as an extension of the ideas used in the VFH* algorithm. The algorithm receives a subsection of the certainty grid with robot's past position at its center as input. The red dot in Fig. 7 shows the actual current position (P') of the robot, while, the blue dot shows the position of the robot for which the CR must be extracted (P).

The first step of our algorithm involves mapping the CR to a polar histogram. The method for generating a polar histogram is same as that mentioned in [1] and takes the CR image with P being the Robot Center Position (RCP). The vector field direction (β_i) and magnitude (m_i) for each pixel of the CR are determined with respect to the RCP in conformance with [1] as follows:

$$\beta_i = \tan^{-1}\left(\frac{y_o - y_i}{x_i - x_o}\right) \quad (1)$$

$$m_i = c_i^2(a - bd_i^2) \quad (2)$$

where,

$x_o, y_o =$ Coordinates of the RCP
$x_i, y_i =$ Coordinates of i^{th} pixel
$c_i =$ Certainty value of i^{th} pixel
$d_i =$ Distance from i^{th} pixel to the RCP

The parameters a and b are chosen according to the following rule in [1]:

$$a - b\left(\frac{w_s - 1}{2}\right)^2 = 1 \quad (3)$$

To be noted is the fact that the distance function (d_i in Eqn. 2) is rotationally symmetric with respect to the RCP. Thus, there is no biasing on any bin/sector due to robot pose. Finally, we generate a polar histogram by adding each pixel's vector magnitude to the bin corresponding to its vector direction with respect to the RCP. Each bin of this polar histogram represents a sector and its total value (ξ) represents the obstacle density in the corresponding direction.

The larger the value of total obstacle certainty (ξ_i) in a bin/sector the more likely it is an obstacle. Based on this observation, we construct a binary polar histogram. Each sector is classified as either obstacle, free or uncertain on the basis of the following equations:

$$\gamma_{i,t} = \begin{cases} 1, & \text{if } \xi_{i,t} > \tau_{high} \\ 0, & \text{if } \xi_{i,t} < \tau_{low} \\ \gamma_{i,t-1}, & \text{otherwise} \end{cases} \quad (4)$$

where,

$\gamma_{i,t} =$ Binary Value for bin i at time t ($1 \implies$ blocked)
$\xi_{i,t} =$ Value of polar histogram in bin i at time t
$\tau_{high} =$ Polar Histogram high threshold
$\tau_{low} =$ Polar Histogram low threshold

Gaps in the polar histogram are defined as sets of consecutive sectors that are 0 in the binary polar histogram. Only gaps with size i.e. angular span (ϕ) above a certain threshold (ϕ_{thresh}) are finally selected as gaps that will be tracked. This ensures that only if a gap has a width comparable to a road will it be tracked, ruling out small off-road paths.

The direction of a gap over time changes as the robot moves forward. Provided that the gap observations are made at consecutive certainty grid positions, it can be proven that the maximum possible positive change in gap direction

Fig. 8. The diagram of the 2^{nd} proposed intersection detection algorithm.

(a) Input Image (b) Wavelet Transform (c) Hough Transform (d) Skeleton

Fig. 9. The sub-figures show the steps of the 2nd algorithm. As can be seen the wavelet transform effectively gets rid of the most of the noise while maintaining the edges. In fig. (d) the intersection point is shown in yellow.

between consecutive observations ($\Delta\theta = \theta_{i,t} - \theta_{j,t-1}$) would be approximately equal to maximum of the angular spans of the both gaps ($\max(\phi_{i,t}, \phi_{j,t-1})$). Thus, we define the following condition as the rule for matching two gaps at consecutive positions or time instances:

$$|\theta_{i,t} - \theta_{j,t-1}| <= \max(\phi_{i,t}, \phi_{j,t-1}) \tag{5}$$

where,

$\theta_{i,t}$ = Median angle value of the $i^{th} gap$ at time t
$\theta_{j,t-1}$ = Median angle value of the $j^{th} gap$ at time $t-1$
$\phi_{i,t}$ = Angular span of i^{th} gap at time t
$\phi_{j,t-1}$ = Angular span of j^{th} gap at time $t-1$

The implemented gap tracking is an iterative process that tracks gaps over successive positions a robot moves to or the successive time-instances when the gaps are observed. We use a polar graph as shown in Fig. 6 to keep track of the gaps. It should be noted that where the polar histogram represents the obstacle density across directions, this polar graph represents the directions in which gaps are observed w.r.t. time. (see relation between Fig 7 and 6) The radius of the blue circles in Fig. 6 denote the time or the recency of position. The direction of a gap in the graph is same as its direction in the certainty grid from the RCP. Also, on the radial axis the farther a point from the center, the more recent is the update. The green dots on each of the blue or red circles denote the angles at which the gaps were observed at the corresponding time instant. A link between two vertices indicates that the two vertices represent the same gap at consecutive instances of time i.e. there is a match between them according to Eqn. 5.

Intuitively, one can see that over time (i.e. along the radius in graph) the longer a chain gets, the more certain is that the particular gap is wide enough to be a road because it was seen from a large set of positions. Since a minimum of two gaps would exist for the road itself (forward and backward directions), presence of any extra gaps gives evidence for another branch. If three or more chains extend simultaneously across a sufficiently long time segment (as shown in Fig. 6(d) with red circles), it would denote presence of an intersection near the location of RCPs used for corresponding gap observations. These RCPs form the zone a forking branch of the road is *visible* and hence, these coordinates are marked as an intersection zone. Additional details about the intersection, such as direction(s) of the forking road(s), are derived from the gap directions observed from these locations and used along with the marked coordinates later for further exploration.

III. Alternative two-step approach to Intersection Detection

Our second approach to intersection detection is a 3 step process. First, we do a multi-resolution wavelet based analysis of the certainty grid image. This is followed by a modified hough transform. Finally, we do a skeletonization to determine the intersections in the image.

A. Multi-Resolution Wavelet Analysis

As the certainty grid image represents the fusion of data between two sensors it is inherently noisy. On an uneven road, the representation of the environment obtained from the sensor readings may have a large component of noise to it. Therefore, we need a method that will help recognize the actual edges in the image from the noise. These edges correspond to the boundaries of the navigable regions in the environment.

A Multi-resolution wavelet analysis has several advantages over simple edge detection operations. At a larger scale, wavelet filters are effective at removing noise, at the same time they increase the uncertainty of a region rightly being detected as an obstacle or not. On a smaller scale the obstacle detection is far more accurate but is susceptible to noise. Thus, combining the multiple scales results in a technique that is robust and accurate.

We use a wavelet based MRA because, sensor readings obtained from certain problematic regions in the map, like

mud, may not be recognized as non-navigable and therefore not classified as obstacle.

Fig. 10 shows the flowchart of how MRA works. In order to smooth the image we have to use a smoothing function $\phi(x, y)$. In our case, we use a standard Gaussian smoothing function $\theta(\mathbf{x}, \mathbf{y})$:

$$\theta(x, y) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

For edge detection we use a special class of wavelets, which are the derivatives of the smoothing function, i.e.

$$\psi^1(x, y) = \frac{\partial\theta}{\partial x} = -\frac{x}{2\pi\sigma^4} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

$$\psi^2(x, y) = \frac{\partial\theta}{\partial y} = -\frac{y}{2\pi\sigma^4} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

where $\int\int_{R^2} \phi$(x,y)=1 and . Since the wavelets use a Gaussian as the smoothing function, this class of wavelets is called the Gaussian wavelet.

The scale of a wavelet can be used to detect edges at different levels of scale. The smoothing function used in the construction of a wavelet reduces the effect of noise. What we have in effect is a robust, scale invariant method for extracting the boundaries of a road from a noisy image. Wavelets are used in a similar way in detection of clouds from geo-spatial aerial images that have a high noise content [10].

The value of σ associated with the Gaussian kernel is chosen so that:

- In areas of uniform intensity, its value is large, thereby allowing it to smooth out random noise.
- In areas of intensity change it's value is small, allowing the edges of the image to be preserved.
- It does not change drastically in it's neighborhood, avoiding broken edges.

B. Hough Transform

After, the MRA, we use points uniformly sampled from a curve to fit lines using the hough transform. This usually results in too many spurious images. Thus, we propose a merging condition to merge similarly oriented lines close to each other.

The straight line hough transform [13] involves transforming a pair of points from the normal Cartesian space to a parameter space, where lines are detected through a voting procedure. Associated with the voting procedure are certain parameters, which are set as per requirement.

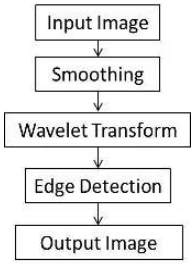


Fig. 10. The process diagram of a multi-resolution analysis.

C. Skeletonization

Our merging procedure is:

Input: The algorithm receives two line segment objects $\mathbf{L_1}$ and $\mathbf{L_2}$

Each line object has 3 attributes: **slope, startPoint, endPoint**

$\mathbf{L}$ = null

if $\|L_1.slope - L_2.slope\| \leq \delta_{slope}$ & $\mathbf{minDist}(L_1.L_2) \leq \delta_{distance}$ **then**

Select **L.startPoint** and **L.endPoint** from the points $\mathbf{L_1}$**.startPoint**, $\mathbf{L_1}$**.endPoint**, $\mathbf{L_2}$**.startPoint**, $\mathbf{L_2}$**.endPoint** such that the length of $\mathbf{L}$ is maximized.

end if

Output: The line object $\mathbf{L}$

---

minDist(L_1, L_2): The minimum distance between L_1 and L_2.

The final stage of our process involves extracting the topological structure (skeleton) of the boundary image. Once extracted, the skeleton contains all the valid topological information required. From the significant branches (branches of more than a certain length) of the skeleton we can extract the location of the intersections. And mark the corresponding stored GPS co-ordinates as intersections.

We use the method proposed in [12] for our skeletonization due to its robustness against noise in the edges. The method uses a significant measure called the bending potential ratio (BPR) which is used to remove spurious skeleton branches.

The BPR is an efficient measure of the way in which contours in the image significant in context to the whole image are considered for skeletonization. The location of a contour is taken into account with respect to the entire object boundary (which in our case is the obstacles in the certainty grid). The BPR effectively integrates both local and global information about the contour. Fig. 9(d). shows the result of the skeletonization after the Hough transform.

We observed that the skeleton branched out at an intersection site only if the gap corresponding to the intersection was sufficiently wide. There is only one parameter, a threshold, that needs to be tuned in order to detect intersections. If the BPR is greater than the threshold then the corresponding contour generates part of the skeleton.

IV. Result

Over the course of several runs we have found that our algorithms have been able to detect intersections with a high degree of certainty. Both algorithms have been used under a variety of lighting and crowd conditions and have proven to be robust and accurate. Two examples of intersection detection are given below. These are followed by the results of a rudimentary exploration.

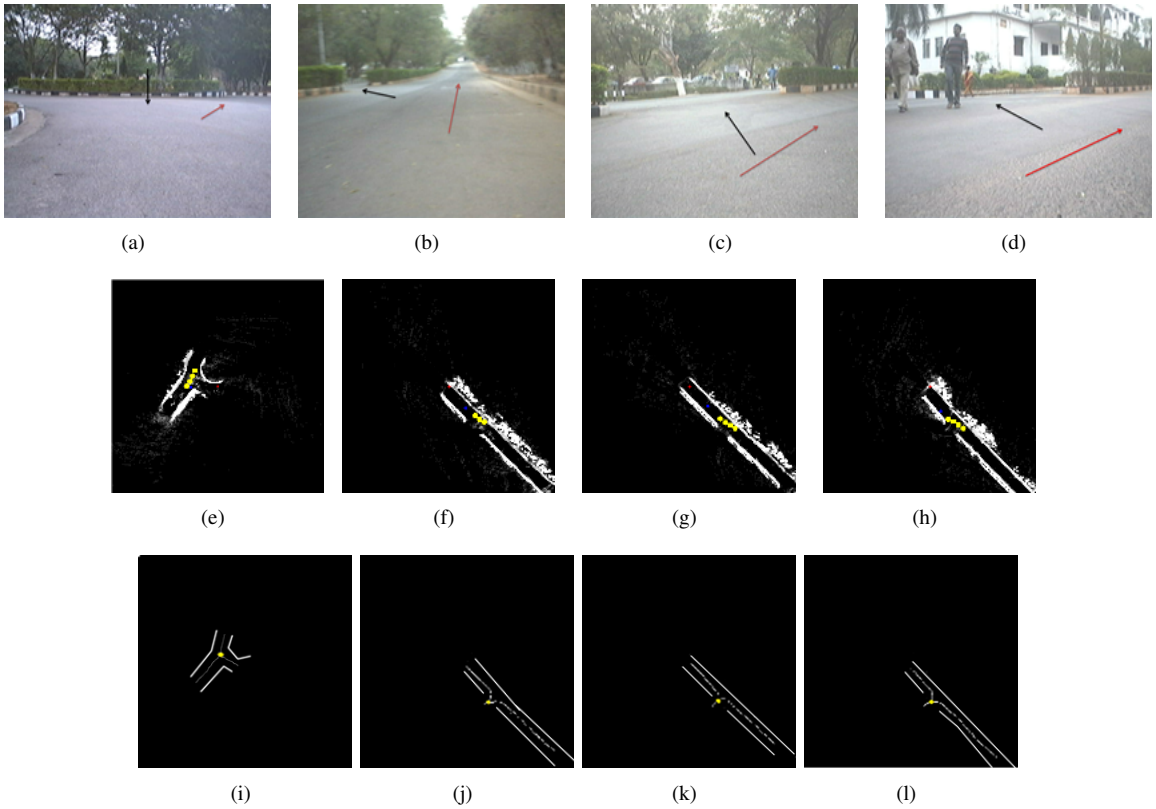
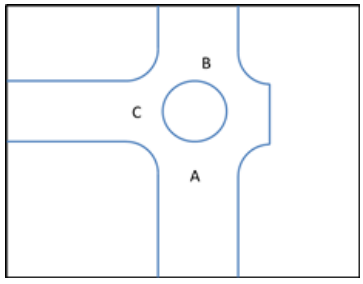
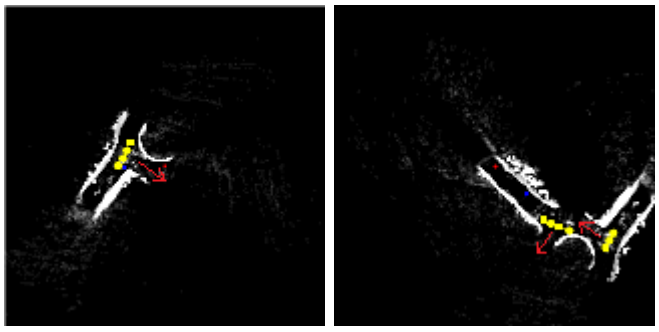


Fig. 11. Top Row: Pictures of intersections. Middle Row: The corresponding intersection detected. by our first method. Bottom Row: The corresponding intersection detected by our second method. The first image is of a roundabout. The black arrows show intersections and the red arrows show the path of the robot.



(a)



(b) (c)

Fig. 12. The roundabout intersection. A top-view. The red arrows in (b) and (c) point to intersections.

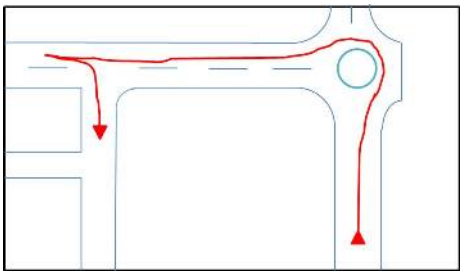


Fig. 13. Path Taken By the Robot During Its Test Run

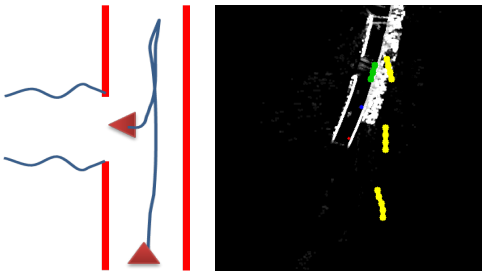
Fig. 12(a) shows a top-view of the roundabout in question. Three roads meet at the roundabout junction, so we have three corresponding intersections. Fig. 12(b) and 12(c) show the certainty grids of the intersections. Our algorithm is able to differentiate and detect all three intersections.

While not encountered during several experiments outdoors, there could be a tendency of the first method to get confused with intersections if the RCP locations for the two intersections overlap. In other words if the robot first detects an intersection, A and while approaching from another direction detect B that is close to A such that their RCP's overlap the intersections could be merged. However, this would not be a problem with the second approach based on wavelets since the skeletonization would detect two distinct branches corresponding to the two forks.

We present the results of an exploration that used the first intersection detection algorithm proposed. The exploration was done using a P3-AT. The P3-AT is equipped with differential wheel encoders, a SICK-LMS laser, an Xsens mti-G inertial unit with GPS and an ordinary webcam. The system was developed in C# using Robotics Developers Studio. The robot has the option of being teleoperated by joystick. This is used as a safety precaution to prevent the robot from crashing.

The robot initially is moving on a straight road, as shown in Fig. 11(a). The first intersection it encounters is at a round-about. The robot takes the path on it's right. In a delayed fashion, this first intersection is detected. The robot follows the curvature of the road, it then detects an intersection to it's right. It goes on straight following the road until the third intersection. The robot then goes on the road extending from the third intersection. As it continues moving along the same road, the robot is able to detect the three intersections, Fig. 11(b),(c),(d) it encounters on it's left. It then continues moving forward upto a point. The robot then goes back to the last intersection and takes the path extending from it. Fig. 11 (e)-(h) show the certainty grids with the detected intersections mentioned in the above paragraph by the first method. Fig. 11 (i)-(l) show the the same intersections by the second method.

In comparison, if a frontier based method was used for exploration and intersection detection, most of the frontiers would correspond to the road curbs. Therefore, some criterion would be needed to choose the frontiers that may be possible intersections. This would have to be done once the frontiers of a particular region have been calculated. Since intersections in an outdoor environment are varied, designing a criterion for selecting the frontiers that correspond to them would be challenging.



(a) (b)

Fig. 14. Fig. (a) shows the path the robot takes when returning to an intersection. Fig. (b) shows the corresponding certainty grid.

A. Recognizing a detected intersection

For exploration it is vital that a robot is able to recognize a previously detected intersection and continue exploring along the fork in the road. Our algorithms have proven robust in recognizing already detected intersections. In this section we show how the robot recognizes an intersection it had previously detected.

Fig. 14 (a) shows a sample path the robot would take in order to detect an intersection. While recognizing an intersection the robot logs the GPS co-ordinates of the intersection. The intersection is also marked in the certainty gird image by a trail of yellow dots. The perpendicular to the trail giving the direction where the road branches out. Once the robot reaches the destination co-ordinates, it turns to face this perpendicular direction. As the robot reaches this intersection it re-detects it. Facing the perpendicular direction it detects road from the sensor information confirming that the current intersection is indeed the previously detected intersection. The recognized intersection is shown by a trail of green dots in Fig. 14(b). The original intersection is shown in yellow adjacent to it.

V. Conclusion

In this paper, we presented two methods for outdoor intersection detection. We have shown extensive results in inhomogeneous conditions on a variety of intersections. We have also shown an application in the form of an exploration. During our search, we were unable to find much literature on the subject of outdoor intersection detection, and believe our methods are novel and have proven robust and accurate in practice.

References

[1] Ulrich, I., Borenstein, J.,*"VFH*: Local Obstacle Avoidance with Look-Ahead Verification"*, IEEE Int. Conf. on Robotics and Automation, April 2000, pp. 2505-2511.

[2] Yamauchi, B. *A frontier-based approach for autonomous exploration.* In Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, July 1997, pp. 146-151.

[3] S. Koenig, C. Tovey, and W. Halliburton. *Greedy mapping of terrain.* In Proceedings of the International Conference on Robotics and Automation, pages 3594-3599. IEEE, 2001.

[4] Krishnan, A., Krishna, M., *A Visual Exploration Algorithm using Semantic Cues that Constructs Image based Hybrid Maps.* In Proc. of Intelligent Robotics and Systems (IROS), 2010.

[5] Meger, D., Rekleitis, I., Dudek, G., *Heuristic Search Planning to Reduce Exploration Uncertainty.* In Proc. of Intelligent Robotics and Systems (IROS), 2008.

[6] Shade, R., Newman, P., *Choosing Where To Go: Complete 3D Exploration With Stereo* In Proc. of the International Conference on Robotics and Automation, pages 3594-3599. IEEE, 2011

[7] Zhang, C., Murai, S., Baltsavias, E., *Road Network Detection by Mathematical Morphology*, In Proc. of ISPRS Workshop, 1999, pp. 185-200.

[8] Geraud, T., Mouret, J., *Fast road network extraction in satellite images using mathematical morphology and Markov random fields* In EURASIP Journal on APplied Signal Processing, Vol. 1, 2004.

[9] Geiger, A., Wojek, C., Urtasun, R., *Joint 3D Estimation of Objects and Scene Layout* in NIPS, 2011.

[10] Hu, J., Razdan, A., Femiani, J., Cui, M., Wonka, P., *Road Network Extraction and Intersection Detection From Aerial Images by Tracking Road Footprints*, In IEEE Transactions on Geoscience and Remote Sensing, Vol. 45, No. 12, 2007.

[11] Thrun, S., Montemerlo, M., et al., *Stanley: The Robot that Won the DARPA Grand Challenge* In the Journal of Field Robotics, 2006.

[12] Shen, W., Bai, X., Hu, R., Wang, H., Latecki, L., *Skeleton growing and pruning with bending potential ratio.* Pattern Recognition 44(2011), 196-209

[13] Pao, D., Li, Hon., Jayakumar, R., *Shape Recognition Using the Straight Line Hough Transform: Theory and Generalization*, PAMI, Vol.14, No.11, 1992.