PASSIVE SKY ANGLE MAPPING FOR UNMANNED GROUND VEHICLES

Bob Grabowski
Richard Weatherly
Robert Bolling
Keven Ring
Chris Scrapper

The MITRE Corporation
7515 Colshire Dr
Mclean, Virginia, 22102

ABSTRACT

We utilize a hemispherical camera to develop a passive mapping algorithm that exploits visibility to the sky. We use this visibility to derive a measure of occupancy to build a wide area volumetric map. Because this method visually exploits the energy of the sky, its effective range can be much higher than an active energy sensor. Since it is passive (it does not emit energy) it is ideal for missions that require stealth. We present initial results from simulation and testing of this new concept.

1 PASSIVE SKY ANGLE MAPPING

Mapping is a critical capability for unmanned ground vehicles. Robots need to know their environment in order to maneuver and navigate in complex terrain. Often they need to evaluate this map to know where to move to minimize danger or maximize sensor coverage. In some cases, these missions require stealth and have to be done at night. However, most mapping sensors today require the emission of energy making them easier to detect. For example, an observer wearing night goggles can easily detect a vehicle utilizing infrared laser range scanners. Another disadvantage of energy emitters is they have to emit a signal that is strong enough to travel to a target, reflect energy and return requiring greater energy demand and limiting their effective range, typically less than 100m.

In this paper, we present a new sensing modality that utilizes a hemispherical camera's visibility of the sky to derive a notion of occupancy. By smartly moving an unmanned ground vehicle, it can exploit this notion of occupancy to passively produce a long range, volumetric map of the environment.

This report is organized as follows. Section II discusses the basic premise of converting a sky angle into occupancy. Section 3 discusses related work in the field of sky detection and exploitation of video to make maps. Section 4 discusses how we represent the volumetric map. Section 5 discusses the need to rectify the raw images to generate a hemispherical image. Section 6 discusses how to segment the sky and calculate sky angle from vertical scans of the image. Section 7 discusses how we sweep the map given a sky angle. Section 8 and Section 9 discusses the experimental setup and presents preliminary results.

2 SKY ANGLE CONCEPT

The sky angle concept exploits visibility to the sky to generate a notion of occupancy. We observe that wherever the sky is visible to an observer (in this case the camera), all points between the viewer and the sky must be free of obstacles. Figure 1 shows a side view representation of this idea with the camera in

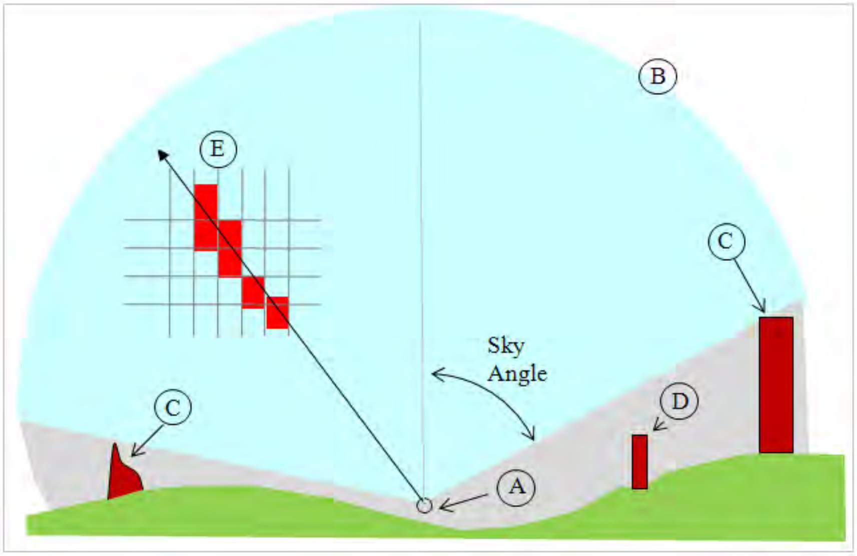

Figure 1 – Cutaway view of objects against the sky: The basic concept of the algorithm is to characterize visibility to open sky. A) The cameras view is characterized as a sphere centered about the camera. B) The open portion of the sky defined by light rays that have reached the camera or conversely emanate from the camera to the sky. C) The highest object in the sky (along a given azimuth) defines the sky angle. D) Objects that fall within the shadow of a taller object cannot be resolved. E) Mapping is accomplished by dividing the sky into equally spaced grids and marking those that the rays have passed through as open or unoccupied.

the center of the cutaway (Figure 1-A). The shaded region above the camera represents the contiguous area where the sky is visible by the camera (Figure 1-B). We then consider a series of rays (line segments) starting from the camera (bottom center) and extending outward towards the sky (outer edge of the circle). Those rays that do not intersect an obstacle have visibility to the sky (bright shaded area). Those rays that intersect obstacles do not have visibility to the sky (dim shaded areas). To understand occupancy for this slice, we divide the area into equally spaced grids (Figure 1-E). Each grid cell that the ray passes through can be characterized as open and can be updated with low occupancy. Rays that intersect objects do not have visibility. For these cells, we cannot infer their occupancy and leave the corresponding cells as unknown.

Figure 2 shows a typical image generated by a hemispherical camera mounted to the top of a vehicle. The hemispherical image has been remapped to a rectangular structure where the top horizontal line represents the very top of the hemisphere and each vertical column of pixels represents the view along a given direction. The hood of the vehicle can be seen in the center bottom of the image. The front of the vehicle is off center to the left (always to the left of the vehicle). The two antennas mounted to the back of the vehicle can be seen on the right. Here the vehicle is passing to the left of a building. The extreme wide angle of the camera produces the distortion of the building.

This concept of sweeping the sky with rays can be extended to a three dimensional volume by casting rays in both azimuth and elevation. In this case, cells represent a volume instead of an area. To understand how to generate a map from determining sky angle, we first look at a simple example composed of six tall cones and a series of stacked boxes. Figure 3A shows the typical result after a single sweep. In this case, we represent open cells as transparent and unknown cells as solid. Cells where the scan rays had visibility to the sky were set to transparent. The resulting pattern looks similar to an irregular lathe applied to a solid. The irregularities of the cut are a result of ray intersection by objects of different heights silhouetted

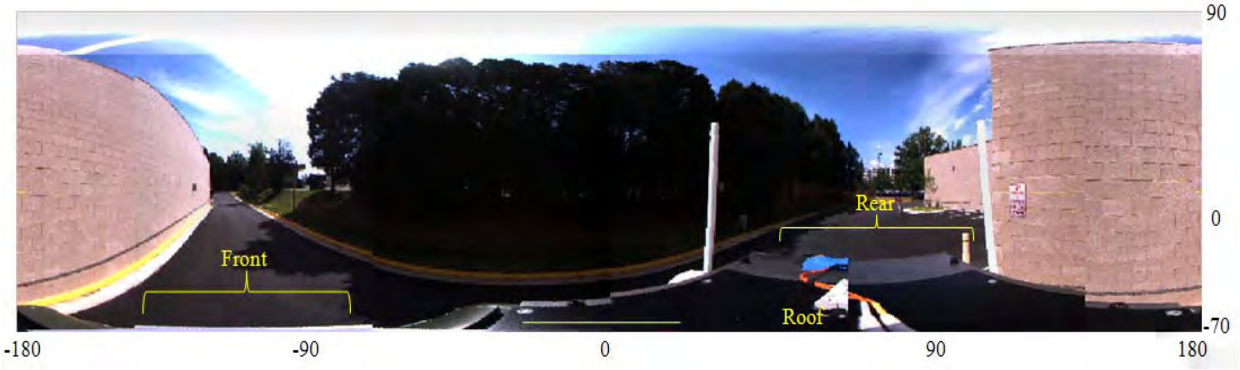

Figure 2 – A typical view from stitching together the six Ladybug images into a single hemispherical camera. The x axis is azimuth and covers 360 degrees around the robot. The y axis ranges from 90 degrees off horizontal (where all the points in the top row map to a single point directly above the robot), to a downward angle of 70 degrees below the horizon. This hemispherical image acts like a very wide angle lens producing distortion of the image.

against the background. Clearly a single reading tells very little about the environment. However, if the same process is executed from several different perspectives, the boundaries of obstacles start to resolve. Figure 3B shows the three dimensional rendering of the environment after 24 scans taken along the dotted path shown. Objects that were viewed from multiple disparate angles show good resolution. Objects that stayed within the shadow of taller objects do not resolve well. Figure 3C shows the final map after an exhaustive sampling of the space. The grid size for this simulation is 500x500 cells.

3 RELATED WORK

Utilizing the energy of the sky to aid in sensing is not new. Moe (Moe 2000), Mckee (Mckee 2005) and Ettinger (Ettinger 2002) uses the sky to aid in determining orientation for UAVs. They segment the sky from the background to develop a rough approximation of the horizon. At even modest heights, objects sticking up above the horizon are negligible and can be ignored. Li describes a method of using the sky with a panoramic image to aid in determining head orientation (Li 2005). He utilizes a gravity sensor to detect pitch and role and then compares the Fourier transforms of successive image frames to measure the phase shift, or yaw between them.

Use of silhouettes to develop models is not new either. Projection of object silhouettes is a common technique in computer graphics for generating curved shadows on curved surfaces (Sen 2003). These projections are used to generate a shadow map which is used during rendering to denote whether a rendered pixel is in the open or in a shadow. In essence, we are looking at the inverse. That is we are trying to determine the object that produced the shadow. Cozman describes a method for comparing features from silhouettes of far mountains to determine orientation for space vehicles (Cozman 1996) Successive images are compared to find the implied rotation between them. This technique relies on the observation that the terrain is relatively sparse with few obstacles impeding view of the mountain ranges. Ramalingam describes a method of localization that uses a silhouette image (Ramalingam 2009). He generates an upward facing hemispherical image essentially looking for the silhouette generated by the buildings. He then matches it against a database to determine the location of a robot in a city. Wong describes a technique of utilizing image silhouettes to develop a 3D model (Wong 2002). This technique requires that images be taken in roughly from a circular path around the object. He then develops a 3D model of an object derived from a monocular camera using shape from motion (Pan 2009). However, this method requires the object to move in front of a stationary camera. Pollefeys develops a 3D terrain model by detecting and matching surface planes from images taken at different viewpoints (Pollefeys 2008). This approach pro-

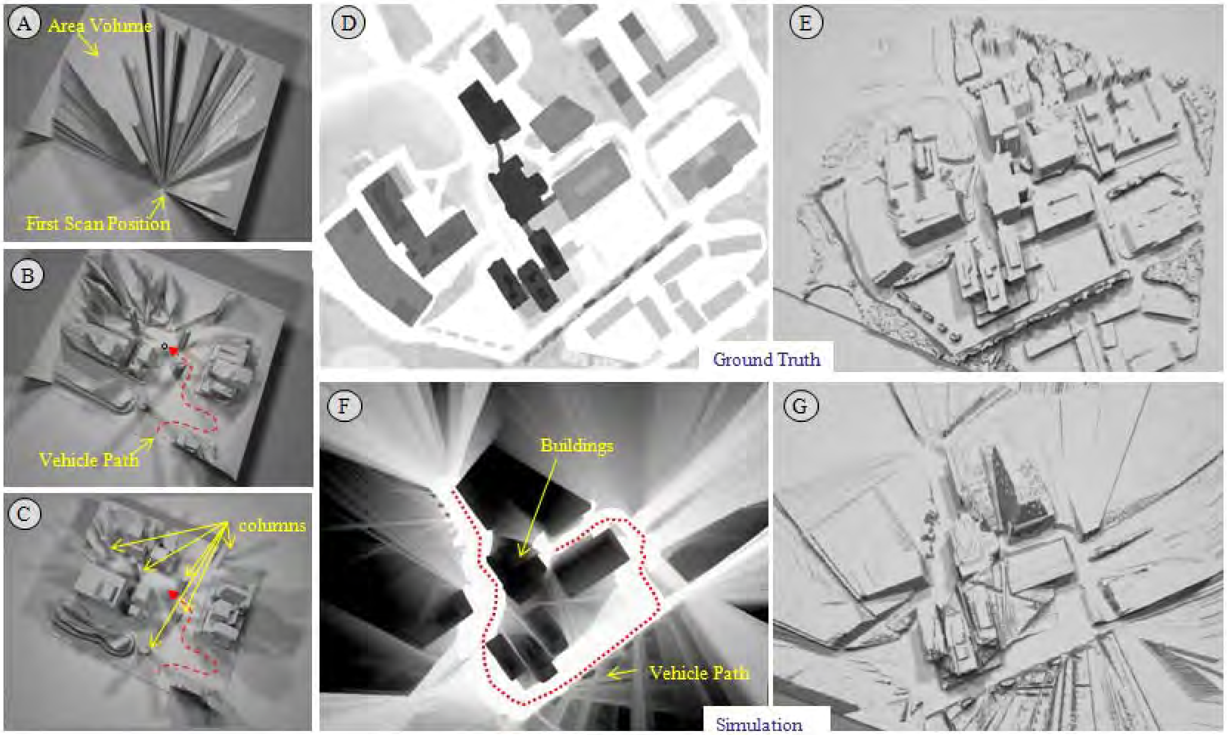

Figure 3 – A) A single scan carves out a volume of open space similar to a lathe where the depth of the cut is based on the heights of the objects along each radial fan. A single scan gives very little information about the structure of the environment. B) The result of the map after moving along the specified path after 24 scans. Places where objects have been viewed from different angles are well defined such as the three columns in the center. C) The final map after moving exhaustively about the space. D) A more complex simulation where the three dimensional environment is represented as a two dimensional heightmap. The intensity of each cell is a function of its height - the darker the cell, the greater its height. White represents a height of zero. E) Rendering of the same heightmap in POVRay, a commercial ray tracing program. The ray tracing algorithm extrudes the height of each cell based on its intensity. F) A map generated in simulation by measuring the sky angle associated with the ground truth from 200 points along a simulated vehicle path. G) The heightmap rendered by the ray tracer. Here the buildings are starting to resolve.

duces high resolution 3D models of the environment. However, it requires large amounts of storage and processing both in the detection phase as well as the constructed 3D models.

4 VOLUMETRIC REPRESENTATION

Given the potential long-range capability of this mapping method, modeling every sub-region (voxel) in a large space can be very expensive in both memory and processing load. To reduce complexity, we adopt a heightmap (also called an elevation map) representation of the environment. A heightmap is a two dimensional structure that represents a three dimensional space and can be stored as an image. The pixels of the image are arranged in equally spaced grids that correspond to equally sized areas in space. The intensity of each pixel represents the height of the object in that grid. The resolution of the image is a function of the size of the image and the area being covered.

Since a heightmap is essentially an image, it is easy to render and display during development providing feedback on how the algorithm is developing. Moreover, heightmaps are graphics objects that can be rendered in most ray tracing software packages. The intensity of each cell is interpreted as a height and extruded in the z direction to produce a three dimension structure. One limitation of a heightmap is that it can only represent the tallest object in that grid space. Consequently, objects with internal gaps, such as bridges or trees, are treated as solid objects. Further experimentation will be necessary to determine if this is an over limiting restriction.

Figure 3D shows a more complex heightmap that represents the MITRE campus modeling buildings, walls, curbs and trees. The white areas are the roads and parking lot around the building and are essentially at ground level. The buildings are the polygonal shapes whose darkness indicates their height. The gray areas represent trees and ground terrain. Figure 3E shows the same heightmap rendered in a POVRay, an open source ray tracing program.

Figure 3F shows the results of simulating a ground vehicle moving through the ground-truth heightmap of Figure 3E along the path indicated by the red dotted line. The results of this simulation is also a heightmap. For each sample along the path, the simulation derives the sky angles to the highest objects and uses it to sweep out open areas in the map volume. For this simulation, a total of 234 samples were made. Buildings in the image that have been viewed from distinct poses (those in the center) have starting to resolve. Places where the buildings were only seen from limited direction (such as the buildings on the outside of the loop), remain ambiguous. Figure 3G shows the 3D rendering of the heightmap generated in Figure 3F.

5 IMAGE RECTIFICATION

The hemispherical image is generated by stitching together the six individual camera images from the Ladybug. However, before we can stitch the images together, we must first correct for the distortion asso-

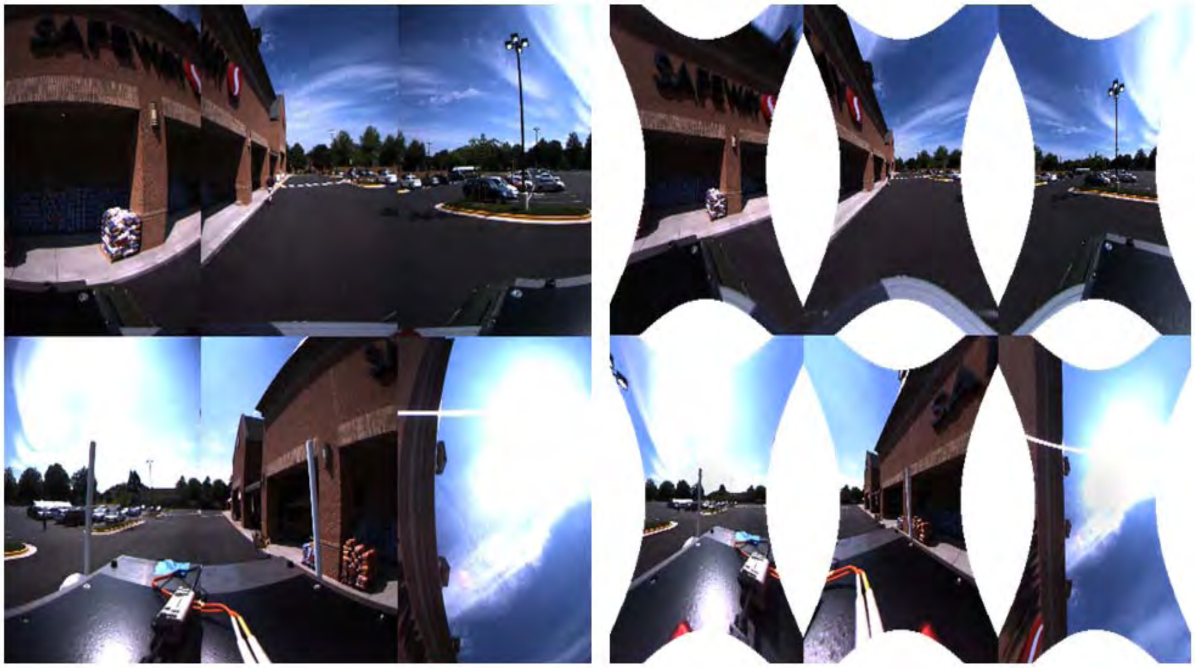

Figure 4 – Image Rectification: A) The six raw images from the Ladybug. Five cameras are spaced equally around the perimeter of the sensor with the sixth one pointing straight up. The images overlap each other slightly. The large field of view introduces radial distortion that produces noticeable barrel distortion. B) Each image is corrected for radial distortion. Note that straight edges such as the light pole and building features are now straight again.

ciated with using wide angle lens. The result of this distortion warps the edges of straight objects such as light poles and building edges (Figure 4a). We make the assumption that the primary contributor to distortion is radial and can be corrected by modeling the distortion using a Taylor expansion along the radial direction:

r_eq=k_1 r+k_2 r^2+k_3 r^3+k_4 r^4

where k1 = 2.2, k2 = -0.005, k3 = 0.0, and k4 = 0.000000010. We also assume the center of the distortion is matched to the center of the digital image. The resultant radius, r_eq, indicates the shift for each pixel along the radial axis. Figure 4b shows the result of this image correction. Note that the straight features, such as the light pole and building edges are now straight again.

Wide angle lens also produce distortion of intensity along the radial direction (Ikeda 2003). That is, the intensity of a constant color source such as the sky drops off quickly near the edges. This results in observable seams between images when stitched. Since we are looking for boundaries between the sky and horizon, this drop off can produce false edges resulting in erroneous matches especially between the top and side images.

6 SKY SEGMENTATION

The first step in determining the sky angle is to segment the sky from objects on the horizon in order to find the highest object for a given viewing direction. We observe that most of our tests are conducted on relatively bright sunny days where the sky is predominately blue with sporadic cloud cover. For our early tests, we chose to detect the sky based on a combination of hue, intensity and saturation.

It is recognized that we chose a simplified method for detecting the sky that is sensitive to illumina-

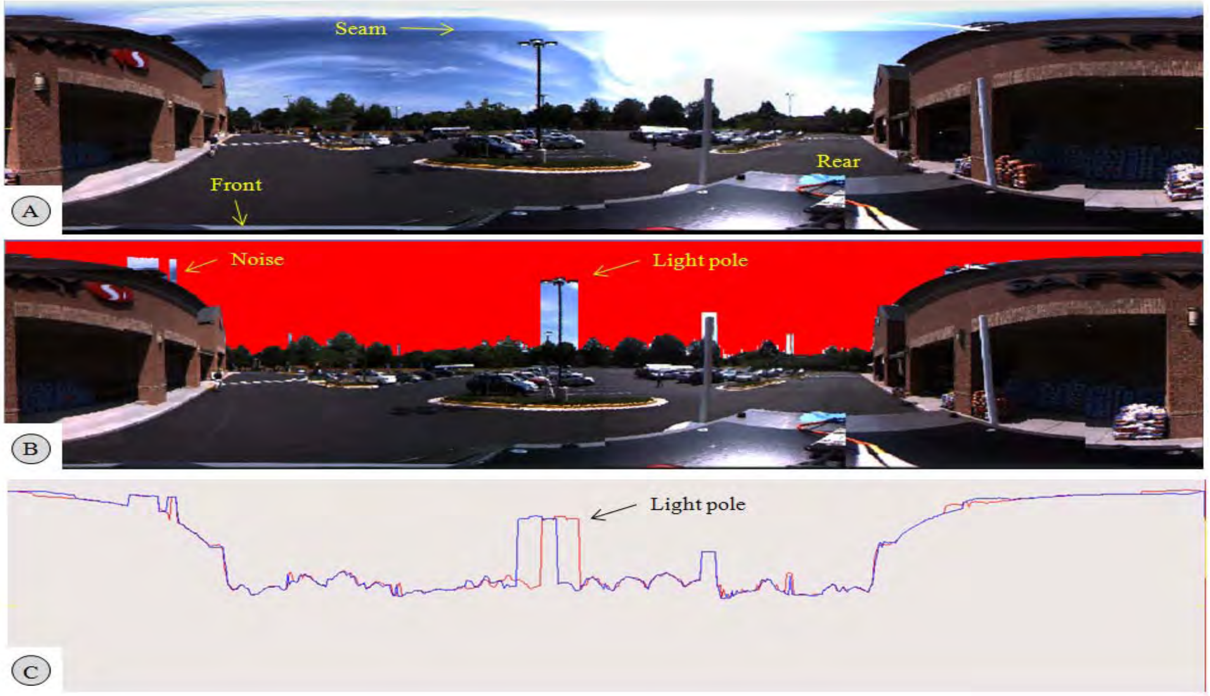

Figure 5 – Sky Segmentation: A) The raw hemispherical image generated from stitching the six Ladybug images together. B) A mask applied to the raw image starting at the top and scanning down until the intensity and hue fall outside the range of the sky model C) Each column scan is stored in a simple array representing the sky angle. The result is a silhouette of the sky with larger objects poking up above the horizon. Here two concurrent images are plotted. Note that objects close to the robot diverge more quickly than objects near the horizon.

tion and coloring effects. More complex approaches, such as (Zafarifar 2004) build complicated models but are still sensitive to proper coloring of the sky. We are exploring the possibility of using infrared to detect sky boundaries. Early results show the sky is relatively cold during both day and night in the far infrared spectrum (10u) and is not susceptible to the lighting effects that plague optical cameras. However, these cameras have a relatively narrow field and do not provide sufficient coverage of the sky. However, the current color based approach produces adequate results to prove the concept.

Sky segmentation is accomplished by scanning from top to bottom of each column of the stitched image. Since the image is a projection of a spherical image onto a rectangular image, each column represents all the pixels that fall along a given azimuth (Figure 5A). Scanning from top to bottom in the image is the equivalent of scanning the sky from directly above the robot down along a straight arc to the bottom of the robot.

We scan downward from the top of a column evaluating each pixel until we find the first non-sky pixel. This first non-sky pixel represents the highest object along that angle. We represent each pixel in terms of its hue, saturation and brightness (HSB space) where each property ranges from 0 to 1. A pixel is considered sky if it is either blue (0.5 < hue < 0.65) or saturated (saturation < 0.2). The saturation component is needed to prevent termination inside a cloud where the color is some shade of white (hue is indeterminate) but has very low saturation (Figure 5A top center). A second termination criterion is added based on intensity (intensity < 0.5) to aid in detection of edges between sky and blue shaded building edges. When a non-sky pixel is found, we terminate the scan and record the elevation angle for that azimuth. This is the sky angle.

Figure 5B shows the result of masking each detected sky pixel in red based on the above criterion. Note that objects, such as the light pole, mask those pixels below it. This visualization also shows a place where the algorithm erroneously terminates in the sky or below the horizon. Figure 5B is typical of most of the images in our data runs. However, there are still a handful of images that erroneously terminate high in the sky or cut into building edges introducing noise in the final maps. We are currently refining our sky segmentation process.

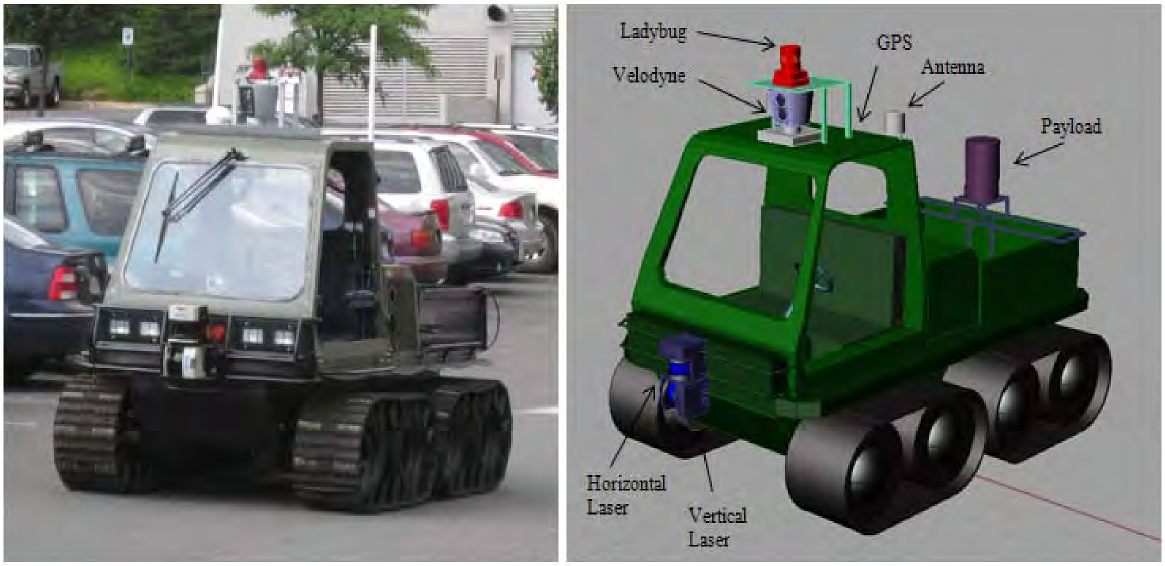

Figure 6 – Setup: Experiments were run using on an autonomous unmanned vehicle called the Centaur. The vehicle platform is equipped with a GPS for position control, two SICK lasers in front provide reactive obstacle avoidance, a 3D laser (Velodyne) mounted to the roof to provide area coverage and terrain understanding and a hemispherical camera (Ladybug) mounted directly above the Velodyne.

Figure 5C shows the resultant sky angle for each bearing and represents the silhouette of the horizon. These sky angles are stored in an array and ultimately passed to the map sweeping algorithm. Figure 5C also shows the sky angle array from the previous scan. Note that close objects sticking up above the horizon, such as the light pole, have translated much more between samples than objects in the background.

7 VOLUMETRIC SWEEP

The final step of the mapping process is to sweep the occupancy map outward from the sensor using the sky angle results extracted from the hemispherical image. The sweep algorithm updates the height of each cell as it climbs away from the sensor at a rate determined by the sky angle.

To ensure each cell of the discrete map is updated, we define an edge array that records the coordinates for each pixel along the edges of the occupancy map. If we simply chose to sweep lines based on the number of vertical columns of the hemispherical image, we would not be guaranteed that the ray trace would touch all cells in the occupancy map.

We begin by determining the starting pixel for the sweep that represents the current sensor location (vehicle position plus sensor offset). We then calculate rays between the starting pixel and each edge pixel in the map. This ensures that after a single sweep, each pixel in the image has been updated. We calculate the rise in z by interpolating the sky angle for the given azimuth. We then sweep along the length of the ray updating the height of each touched cell if necessary. To see if a cell is to be updated, we compare the current height of the cell along the line with the corresponding height of the ray at that cell. If the ray’s height is lower, then that cell is updated with the lower height. If the ray’s height is lower, then the cell is left alone. This process is repeated for each ray.

8 EXPERIMENT SETUP

To generate the panoramic image, we employ the Ladybug camera unit developed by Point Grey Inc. This unit contains six high quality fire wire cameras in a single unit. Five cameras are equally spaced in a ring around the body with the sixth located on the top pointing up. Each of the cameras feed their raw images into a hardware compressor box also provided by Point Grey through an optical link. The compressor reduces the bandwidth needed to pass the images. The compressor connects to a PC via an 800 Firewire link. The compressor allows adjustments of parameters such as image size and frame rate. For these experiments, each image is 384x512 at a frame rate of 2fps**.** The processed camera images are feed into a mini MAC processor running a Java-based agent architecture (Grabowski 2007).

The location for each image is obtained using a GPS module that produces latitude and longitude at 10 times per second. Position measurements and image acquisition are asynchronous. We assume for early testing that vehicles speeds will be relatively small and the mismatch between acquisition and position are negligible.

The camera and GPS unit were mounted on the top of an autonomous robot platform developed at the MITRE Corporation (Figure 6). The robot is a retrofitted four-tracked vehicle called the Centaur which was originally designed for manned oil exploration and remote hunting. Servo controllers have been added to steering, throttle and brake to allow drive by wire capability. Two forward mounted SICK lasers provide rapid and reactive obstacle avoidance. One laser is mounted horizontally providing a 180 degree planar sweep directly in front of the robot. A second vertically mounted laser characterizes the terrain directly in front of the vehicle to provide a rudimentary range gate for the horizontal laser to account for rolling terrain. A gimbaled camera is mounted inside the cab to provide a video feed for teleoperation and other mission specific video. A scanning Velodyne laser is mounted to the top of the cab to provide wide area coverage. Mounted directly above the Velodyne is the Ladybug camera. The Ladybug is mounted directly above the rotation axis of the Velodyne to provide easier fusion between the two. The GPS and radio antennas are mounted just behind and below the roof sensors.

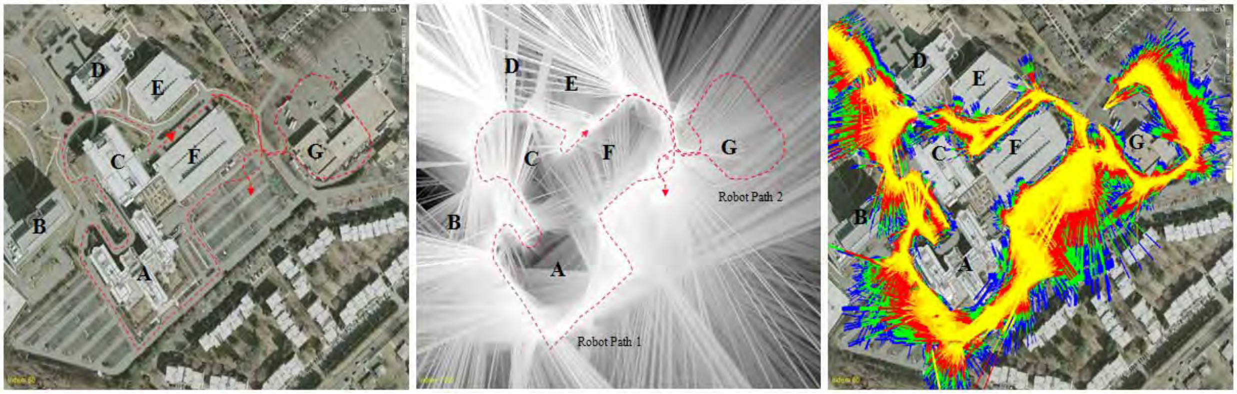

Figure 7 – Results: A) Data was collected from driving among a set of seven buildings on the MITRE campus. Building G was actually torn down and rebuilt in a different configuration (as evident from the resultant maps). The route of the vehicle is shown as red dotted lines. B) The resultant height map for the two data runs. White represents low height, dark represents objects that stuck out high above the horizon. Gaps appear in the map on the north side of building A and building G due to loss of GPS. C) The same map color coded by height. Yellow is lower height. Closer examination between building E and F as well as the east side of building A show evidence of the trees surrounding the site.

9 EXPERIMENT RESULTS

To conduct the experiment, we drove the robot around the MITRE campus along two paths shown in Figure 7A. The total distance travelled was 1774m. A total of 1765 samples were used. Samples were rejected in several places were GPS was lost (just north of buildings A and G).

We have chosen a heightmap image size of 1050x850 that covers an area of 504x407m which represents approximately 0.5m resolution. We consider the maximum height of buildings at 30m and use an image depth of 8 bits (255 intensities) giving a height resolution of 12 cm in the vertical direction.

Figure 7B shows the resultant heightmap generated from the combination of the two data runs. The outlines of the buildings are starting to resolve. Noise in the sky detection process accounts for some of the erroneous cuts across the buildings. Figure 7C shows a highlighted version of the heightmap where the pixels are colored by height to draw out some of the detail.

10 CONCLUSION

In this paper, we have presented a passive mapping algorithm that utilizes the energy of the sky to infer occupancy of a volumetric map. We have shown that by sampling the space around objects, we can resolve details about tall objects in the scene. We have shown some initial simulation results and developed the algorithms for building a volumetric map.

11 REFERENCES

Cozman, F., and Krotkov, E., “Position Estimation from Outdoor Visual Landmarks for Teleoperation of Lunar Rovers,” In Proc. Third IEEE Workshop on Applications of Computer Vision., 1996. pages 156-161.

Ettinger, S., Nechyba, M., Ifju, P., and Waszak, M., “Towards Flight Autonomy: Vision-Based Horizon Detection for Micro Air Vehicles,” 2002 Florida Conference on Recent Advances in Robotics, Miami, May 2002.

Grabowski, R., Weatherly, R., Bolling, R., Seidel, D., Shadid, M., and Jones, A., “MITRE Meteor: An Off-Road Autonomous Vehicle for DARPA’s Grand Challenge,” Journal of Robotics: Special issue on Grand Challenge, Vol 36, 2007.

Ikeda, S., Sato, T., and Yokoya, N.,“A Calibration Method for an Omnidirectional Multi-Camera System,” Proc. SPIE, vol. 5006, pp. 499-507, 2003.

Li, S., "Estimating head orientation based upon sky-ground representation," Intelligent Robots and Systems, 2005. 2005 IEEE/RSJ International Conference, vol., no., pp. 844-849, 2-6 Aug. 2005.

McGee, T., Sengupta, R., and Hedrick H., “Obstacle Detection for Small Autonomous Aircraft Using Sky Segmentation,” Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp.4679-4684, April 2005.

Moe, A., “Passive Aircraft Altitude Estimation using Computer Vision,” Lic. Thesis LiU-Tek-Lic-2000:43, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden, thesis No. 847, ISBN 91-7219-827-3, Sep. 2000.

Pan, Q., Reitmayr. G., Drummond, T., “ProFORMA: Probabilistic Feature-based On-line Rapid Model Acquisition,” Proc. 20th British Machine Vision Conference, London England, September 2009

Pollefeys, M., Nistér, D., Frahm, J., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R., Welch, G., and Towles. H., Detailed real-time urban 3d reconstruction from video. Intelligent Journal of Computer Vision, 78(2-3):143–167, 2008.

Ramalingam, S., Bouaziz, S., Sturm, P., and Brand M., “Geolocalization Using Skylines from Omni-Images,” Proceedings of the IEEE Workshop on Search in 3D and Video, Kyoto, Japan, Oct. 2009

Sen P., Cammarano M., Hanrahan, P., “Shadow Silhouette Maps,” ACM Transactions on Graphics, Vol. 22, No. 3, July 2003.

Wong K., and Cipolla, R., “Reconstruction of outdoor sculptures from silhouettes under approximate circular motion of an uncalibrated hand-held camera,” in Proc. IAPR Workshop on Machine Vision Applications, Nara, Japan, December 2002, pp. 459–462.

Zafarifar, B., and de With, P., “Blue Sky Detection for Picture Quality Enhancement,” Image Processing, 2004. ICIP '04. 2004 International Conference on, Vol. 4 (2004), pp. 2367-2370 Vol. 4.