

3D Hallway Modeling Using A Single Image

Greg Olmschenk, Zhigang Zhu
The Graduate Center and the City College
The City University of New York

golmschenk@gc.cuny.edu, zhu@cs.cuny.cuny.edu

Abstract

Real-time, low-resource corridor reconstruction using a single consumer grade RGB camera is a powerful tool for allowing a fast, inexpensive solution to indoor mobility of a visually impaired person or a robot. The perspective and known geometry of a corridor is used to extract the important features of the image and create a 3D model from a single image. Multiple 3D models can be combined to increase confidence and provide a global 3D model. This paper presents our results on 3D corridor modeling using single images. First a simple but effective 3D corridor modeling approach is introduced which makes very few assumptions of the camera information. Second, a perspective based Hough transform algorithm is proposed to detect vertical lines in order to determine the edges of the corridor. Finally, issues in real-time implementation on a smartphone are discussed. Experimental results are provided to validate the proposed approach. This work has the potential to also function in environments with properties analogous to corridors such as highways, sidewalks, city blocks, etc.

1. Introduction

In the world built by humans, hallways are everywhere. As such, there are many cases where the information about the corridor is important. For one, robots are continually becoming more common. A mobile robot needs to be able to navigate its environment. If that environment involves the inside of a building, its likely the robot will frequently encounter hallways. Furthermore, hallway detection can be used to help visually impaired people avoid the walls as well as find the doors and cross section corridors necessary to take them to their destination.

The most common way to extract the information on a corridor in computer vision currently is to use a range sensor and perform the computations necessary to find the walls of the corridor. Unfortunately, these methods require relatively expensive equipment, heavy processing, and high energy consumption. This makes the method impractical

for a robot that needs to move quickly, a wearable system that need to be built cheaply, or a blind person who doesnt want to carry a laptop on their back processing data all day. The most recent relatively low-cost 3D sensors are the RGB-D sensors, such as Microsoft Kinect and Asus Xtion Pro, but their sensing ranges are quite limited (0.5 to 4 meters) and are not very suitable for corridor detection. We propose a high speed, low resource, long range, inexpensive solution to these needs. Effective portable corridor detection capabilities with nothing more than a single consumer camera such as a webcam or the cameras found on a smartphone. This is done through the detection, localization and 3D reconstruction of corridors from individual images, while multiple images can provide more detailed and confident information. Furthermore, as the RGB camera does not have a range limitation as the range sensors do, the methods described here can be combined with data already retrieved from range sensors to make assumptions about what the camera can see beyond the range sensors limits.

We have made the following three contributions for enabling a visually impaired people to navigate in a typical corridor scenario. First a simple but effective 3D corridor modeling approach is introduced which makes very few assumptions of the camera information: we only need to assume know camera height and it is without a rolling angle, and the 3D information of the corridor and the floor plan is obtained by utilizing the vanishing point of all the horizontal lines. Second, an efficient perspective based Hough transform (PBHT) algorithm is proposed to rapidly detect vertical lines in order to determine the edges of the corridor. The algorithm efficiently makes full use of the 3D information obtained via vanishing point analysis so vertical lines near and far are all used for determine the edges of the corridor. Finally, issues in real-time implementation on a smartphone are discussed, including model representations, computation costs and communications with blind users. Experimental results are provided to validate the proposed approach.

The rest of paper is organized as follows: First, Section 2

discusses what's been done in the past, how it relates to what we did in our research, and what we've done new. Next, Section 3 gives an overview of the approach used in this paper. Section 4 describes the algorithms used to model the corridors in detail. Then experiment results are shown in Section 5. Finally, conclusions are provided at Section 6.

2. Related Work

The majority of existing corridor detection methods use range sensor data to extract corridor information. The most detailed indoor modeling methods use point cloud data gathered by 3D range sensors [5]. This 3D data can often be processed more quickly by reducing the point clouds to surfaces [12]. Other range methods use 2D laser range sensors to provide a faster, less resource intensive approach [12]. These range sensor methods use powerful hardware to extract precise position data without the need to interpret information based on a RGB image. The most recent low-cost RGB-D sensors, such as Microsoft Kinect and Asus Xtion Pro, can obtain depth maps in real-time, but their sensing ranges are quite limited (0.5 to 5 meters) and are not very suitable for corridor detection and 3D measurements of over 10 meters long. Our approach allows for fast, efficient processing without the need for 3D equipment. Furthermore, our methods can be used in conjunction with a range sensor to provide initial data and make decisions about the environment beyond the range sensors limit, while the range data can be processed slower to give more detailed results.

The use of a single RGB camera to detect doors has been used extensively in existing research [10][14]. The means used to find the frames of the doors in these methods is similar to the extraction of corridors in our approach. In particular, the vanishing point is used to determine a perspective so that important feature lines can be extracted. The major differences between the method given in this paper and the door detection are the means by which the features are found. While these related works use simple edge detection and a search for a specific shape, our method utilizes our PBHT and the characteristics of a hallway to gather important features (such as corridor boundaries as well as vertical lines) in the image. We then generate 3D information from these features using well established methods [2].

Past research has been done on allowing a robot to quickly move through a corridor [9]. However, these past methods rely on existing 3D maps of the corridors which the robot matches its currently surroundings with. The method described in this paper expands on this by requiring no previous knowledge. Other fast corridor detection is done with powerful laser range sensors [8][7] leading to cumbersome equipment being required.

Detection of corridor vanishing points to determine the direction of the hallway has been extensively studied [6]. We build on these methods by utilizing and creating a model

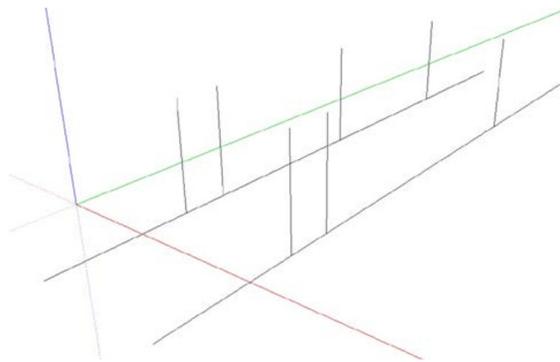


Figure 1. 3D wireframe model of a corridor, with boundaries between the two side walls and the floors, and the frames of doors.

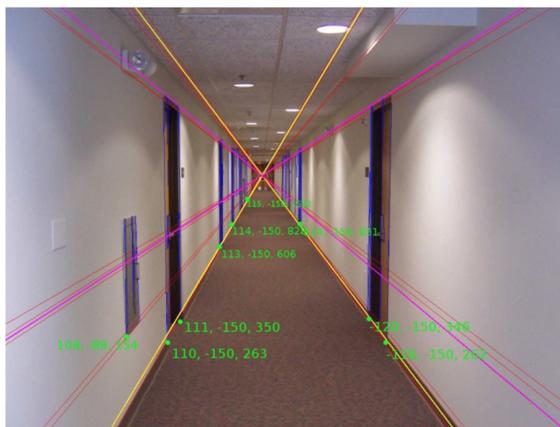


Figure 2. A typical image of a corridor. The direction of the corridor is detected by finding the vanishing point of horizontal lines (in pink), the boundaries of the corridor (the yellow lines below the vanishing point) via the detection of vertical structures (the blue lines). 3D measurements (X,Y,Z) are also labeled for a few vertical lines (in cm).

of the hallway and its features such as doors, turns and corners of the hallways.

Other research has previously used the properties of environments similar to hallways to perform 3D reconstruction [13][1][3][4]. However, our method extends this by using the perspective to improve results at a distance. Even though incorporating perspective into a Hough transform is not a new idea [11], existing approaches attempt a more general solution in which performance suffers. Our method is designed efficiently for the detection and 3D measurements of corridors using a smartphone so that useful results can be used in applications such as the guidance of the visually impaired.

3. Overview Of Our Approach

While most uses of 3D reconstruction require multiple viewpoints, a range sensor, or some other special imaging component, our methods can perform the task using a single image. This is due to the nature of a hallway (Figure 1, Figure 2). Even though there are plenty of differences from one hallway to the next, the feature of having (mostly) straight walls makes the single image processing possible. From the camera's point of view, many of the lines found in an image run to a vanishing point. More specifically, the lines which define the boundaries between the floor to the walls and the walls to the ceiling are lines that go to the vanishing point. Furthermore, lines that perpendicular to the hallway (e.g. doorframes, floor tiles, etc.) will appear as lines parallel to one another that do not go to the vanishing point. To perform the reconstruction, we begin by performing a Canny edge detection. The result is then put through a Hough transformation to produce a map of lines. Taking into account the special vanishing point feature of a corridor, we are able to determine that the position of the vanishing point will be where the most lines from the Hough transformation converge. From here, we search for surface features, parallel lines which do not run towards the vanishing point. This is done using a perspective-based Hough transformation (PBHT) which retrieves line segments. This PBHT is specially built based on the perspective of the image and computationally efficient for smartphone implementations. That is, in the hallway image there may be strongly defined lines (such as doorframes) far down the hallway, yet they would be normally excluded from a regular Hough transform because these lines are so much smaller than lines physically closer to the camera. To account for this, the Hough transform adds a weighting biased towards objects closer to the vanishing point. The line segments found serve two purposes. First, the line segment will lie on one of the surfaces, so the end of this segment can be used to estimate which line that passes through the vanishing point defines the boundary between one surface to the next. Second, the line segment can be used to find features. The most obvious example of this is the line that outlines a door or connecting hallway. Since the number of points and lines used in the algorithms are small, the computation is efficient. From this information, a 3D wireframe model of the hallway can be built.

4. Corridor Modeling Algorithms

We will give describe the mathematical model of a 3D corridor as the foundation of our algorithms in detecting vanishing points and boundaries of the corridor. Then the algorithm for vanishing point detection will be presented, followed by the description of our proposed perspective-based Hough Transform algorithm.

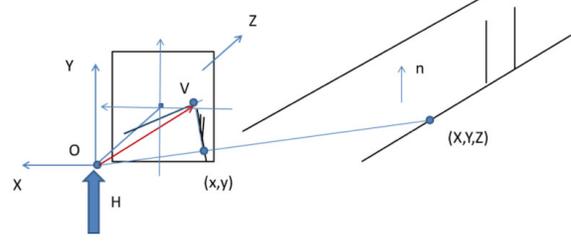


Figure 3. Corridor Detection And Modeling: The Geometry

Once a 3D wireframe model has been generated, the data can be analyzed. Features on the surfaces of the walls, floor, and ceiling can now be compared with relative distances. Particularly, important features such as doorways/openings may be extracted. Furthermore, a set of wireframe data easily combines with another set of wireframe data. Two such sets quickly provide absolute distances and increases the confidence of the data.

4.1. 3D Corridor Modeling

We assume the height of the camera (H) is known (e.g. at the height of a users chest) and the focal length is F . We also assume that the camera is leveled in the horizontal direction (along the X axis), or the rolling angle of the camera can be estimated using the on-board sensor and images rectified, so that we only have a heading angle (α) and a tilt angle (β) relative to the corridor (Figure 3). If the vanishing point of the corridor axis is $v = (x_0, y_0, F)$ representing in the camera coordinate system, then the heading and tilt angles of the camera can be calculated as:

$$\alpha = \text{atan} \left(\frac{x_0}{F} \right), \beta = \text{atan} \left(\frac{y_0}{F} \right) \quad (1)$$

The normal of the ground plane in the camera coordinate system is $\mathbf{n} = (0, F, -y_0)$. Define the magnitude of the normal vector as $n = |\mathbf{n}|$. Since the camera is leveled, then the plane equation can be written as:

$$FY - y_0Z = nH \quad (2)$$

Hence, given a floor image point (x, y) , the 3D coordinates of the corresponding point can be calculated as:

$$(X, Y, Z) = \frac{nH}{F(y - y_0)}(x, y, F) \quad (3)$$

when $y \neq y_0$. Otherwise the ground plane equation would be $Y = H$, and the corresponding 3D point would be simply $\frac{H}{y}(x, y, F)$. From this, the distance of Z each image point on the floor can be estimated. The location and the width of the corridor, and the location/width of each door can be calculated. The width of a door can be calculated as the distance between two endpoints of the door frames on

the floor. The distance of each edge of the corridor can be calculated as the perpendicular distance of the camera center to the 3D line of the edge once the 3D coordinates of two points on the lines are obtained.

4.2. Vanishing Point Detection

First we need to determine the vanishing point of the corridor image. This vanishing point and the characteristics of a corridor shape are essential to generate the 3D model from a single image.

The canny edge detection in our implementation begins with a noise reduction using a 5x5 gaussian filter. Next, a pair of convolution masks are applied in both the x and y directions using the 3x3 Sobel operators. From this a gradient intensity map of the image is generated. A non-maximum suppression is applied which sets all points to zero which are not part of the local maximum. This results in only thin lines (edges) remaining. The canny edge detection terminates after filtering pixels based on thresholds. The detection results in an edge map. Such an edge map is generated for each of red, green, and blue versions of the image individually, then the maps are combined into a single edge map.

The edge map then is then subjected to a standard Hough transformation. Every edge point is used to add to an accumulator array specified by polar coordinates. If each bin of the array which exceeds a given threshold specifies an existing line. The result of the transform is a list of lines given in polar coordinates.

The pixels of the image are then surveyed to find which pixel has the most lines passing through or near it within a given buffer distance. Based on the characteristics of a corridor, this location is determined to be the vanishing point of the image. The vanishing point of the corridor is noted as (x_0, y_0)

4.3. Perspective Based Hough Transformation

Next our method extracts the surface feature lines from the image. To do this we propose a perspective based Hough transformation (PBHT). Most of the Hough Transform approaches including the one that incorporated perspective-invariance proposed in [11] cannot meet the real-time requirements. Our PBHT is a modified method of the progressive probabilistic Hough transformation (PPHT)[13]. The use of PPHT has several advantages over a standard Hough transformation. It is generally faster as it only needs to add a portion of the total points to the accumulator, it results in a finite line segment rather than an infinite line, and it can be stopped early and still present information on the most prominent lines.

In addition to use the efficiency of the PPHT algorithm, the PBHT weights voting values based on the proposed line's normal distance from the vanishing point. This al-

lows for features at a distance to be more correctly recognized as significant despite their small relative pixel size on the image, therefore it integrates both advantages in accuracy and efficiency in a unified algorithm. The algorithm works by first randomly selecting a point to vote. After a vote has been cast, we check the bins to see if the current values would be achieved by random noise. More specifically, from the total value N allocated to all bins, does any bin exceed a threshold of s . The threshold s changes as votes are cast. Once a line is found, the values of the points on that line are removed from the bins and other points supporting that line are removed from the map of points which have not yet voted.

Every pixel votes into exactly one bin for each θ value. This allows us to only require analysis along the ρ dimension of the accumulator. The value allocated to each bin is based on the shortest distance of the line defined by that bin to the vanishing point. The value attributed to the bin based on a line distance is proportional to

$$v = \frac{1}{d + c} \quad (4)$$

where d is the shortest distance of the line from the vanishing point and c is a constant based on the size of the image. Since we are only searching for vertical lines in the PBHT, this is equivalent to using the distance information of these vertical lines: the closer a line to the vanishing point, the larger the distance of the line to the camera. This constant prevents extremely small numbers of pixels close to the vanishing point from being considered lines.

The PBHT algorithm works in detail as follows:

1. If the input image is empty, end the algorithm.
2. Update the accumulator with random pixel from the input image.
3. For each line that passes through the pixel, get the distance of that line to the vanishing point. Update the accumulator with values inversely proportional to the distance.
4. Remove the pixel from the input image.
5. Check if the bin with the highest value of the bins modified by the new point exceeds the threshold. If not, go to step 1.
6. Search the given line for the longest number of continuous pixels which do not have a gap larger than a given threshold.
7. Remove the pixels in the line segment from the input image and subtract the vote values for each bins of those which have already voted.

8. If the line segment total vote value exceeds a given threshold value, add it to the output line list.
9. Go to step 1.

The steps which are significant from a computational complexity point of view are 3 and 9. Step 9 represents the overall loop on each pixel in the image. While step 3 diminishes in time consumption with each loop of step 9, it is still the computational heavy step in the process. These two factors combined results in a $O(n) = n \cdot \log(n)$, where n is proportional to the number of pixels in the image.

Once the surface line segments have been found, the transition from floor to the walls is found based on the end points of the line segments. A simple linear regression of the end points is taken and the line from the original Hough transformation which most closely matches the result is determined to be the transition line.

5. Experimental Results And Discussion

5.1. Experimental Results

In our tests, the center of the image and the focal length are determined simply by using the metadata of the image. The center is simply calculated as the middle of the image and the focal length is converted into pixels using the size of the sensor target.

In our first example (shown in Figure 2), the image center and focal length are $C_{image} = (610px, 457px)$ and $F = 343px$. Based on these parameters we find the vanishing point $V = (53px, 92px)$. Therefore, the heading and tilt angles of the camera can be calculated, using Eq. (1) and Eq. (2), as $\alpha = 4.41^\circ$ and $\beta = 7.64^\circ$. This means the heading angle is almost straight forward and there is only a slight tilt. From here, the normal of the ground is calculated as $\mathbf{n} = (0, 343px, -92px)$.

The calculated 3D position for several of the door features are given in the table below. Note that one of the points is 16 meters away, which is not possible to obtain by a RGB-D sensor or a typical stereovision system with short baseline length.

Left Side			Right Side		
X (cm)	Y (cm)	Z (cm)	X (cm)	Y (cm)	Z (cm)
108	-89	154	-118	-150	262
110	-150	263	-120	-150	346
111	-150	350	-124	-150	831
114	-150	606			
114	-150	828			
115	-150	1622			

In our second example (Figure 4), we have a more complex scene than in the first. Lockers provide many additional features and pillars of unconventional shape affect the data. The image center and focal length are $C_{image} =$

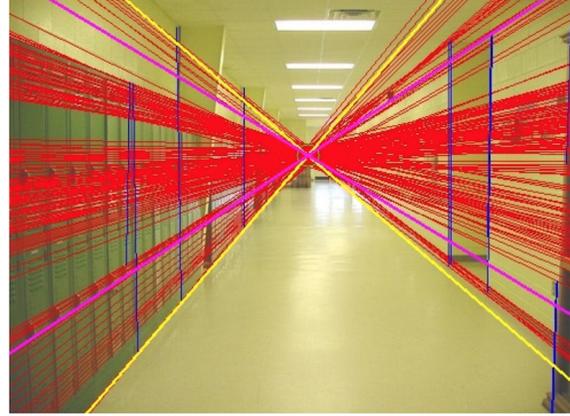


Figure 4. Example 2: A corridor with a bank of lockers. Red lines are all lines which voted for the accepted vanishing point. Blue are the detected vertical line features. Yellow are the transitions from the floor to the walls. Purple are from the walls to the ceiling.

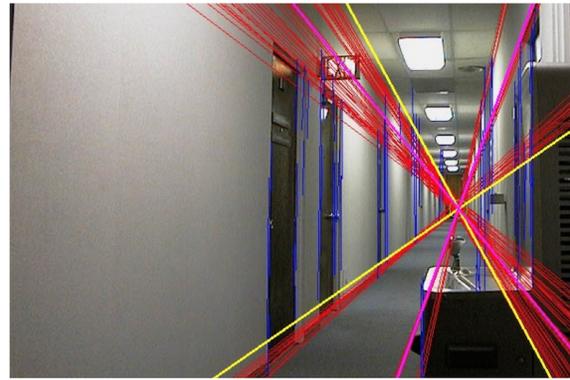


Figure 5. Example 3: Heading to one side of a corridor with a large obstacle. The color of the lines have the same meaning as in Figure 4.

$(246px, 185px)$ and $F = 181px$. Based on these parameters we find the vanishing point $V = (-15px, 47px)$. Therefore, the heading and tilt angles of the camera can be calculated, using Eq. (1) and Eq. (2), as $\alpha = 2.37^\circ$ and $\beta = 7.40^\circ$. This means the heading angle is almost straight forward and there is only a slight tilt. From here, the normal of the ground is calculated as $\mathbf{n} = (0, 181px, -47px)$.

The calculated 3D position for several of the door features are performed, but the results will be omitted from this point on for the sake of space.

In our third example (Figure 5), we have much of the hallway obscured by a water fountain. Here is a situation where multiple images/wireframe models combined would help determine the correct information. The image center and focal length are $C_{image} = (300px, 200px)$ and $F = 269px$. Based on these parameters we find the vanishing point $V = (-178px, -18px)$. Therefore, the heading and



Figure 6. Example 4: A hallway with many students standing.

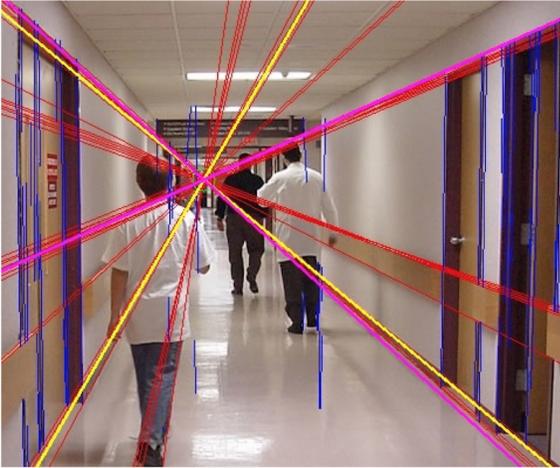


Figure 7. Example 5: A hallway with a few people walking and strange wall features.

tilt angles of the camera can be calculated, using Eq. (1) and Eq. (2), as $\alpha = 18.31^\circ$ and $\beta = 1.92^\circ$. This means the heading angle is far off center and there is almost no tilt. From here, the normal of the ground is calculated as $\mathbf{n} = (0, 269px, -18px)$.

In our fourth example (Figure 6), we have a hallway with many students congregated. The image center and focal length are $C_{image} = (512px, 342px)$ and $F = 311px$. Based on these parameters we find the vanishing point $V = (-56px, 4px)$. Therefore, the heading and tilt angles of the camera can be calculated, using Eq. (1) and Eq. (2), as $\alpha = 5.14^\circ$ and $\beta = 0.37^\circ$.

In our fifth example (Figure 7), we have a few people in the hallway with some strange features on the walls. The image center and focal length are $C_{image} = (241px, 213px)$ and $F = 152px$. Based on these parameters we find the vanishing point $V = (68px, 59px)$. Therefore, the heading and tilt angles of the camera can be calculated, using Eq. (1) and Eq. (2), as $\alpha = 12.61^\circ$ and $\beta = 10.98^\circ$.

5.2. Discussions on Time Performance and Integration

We have generated 3D wireframe models of the corridor from a single image. The wireframe models will be matched based on the relative positions of the prominent feature lines such as the edges of doorframes. Because the geometric characteristics of a corridor are simple, the matching of these 3D models requires very little processing. The major computation cost is in 3D modeling (including edge detection, two rounds of Hough transformation, for vanishing point detection and for corridor/door detection, and 3D measurements). Using images from a video taken by an iPhone, our current implementation on a laptop using C++ has an average processing time of 110 milliseconds. This uses reduced resolution (640x360) images for faster processing. The memory usage of the program being run on the laptop has been limited to match that of the iPhone 5 (1.0 GB). The laptop has a processor whose speed (2.4 GHz) is twice that of the iPhone 5s (1.2 GHz). From this it is reasonable to assume that an iPhone implementation would take approximately twice as long to process an image (200 milliseconds). At the writing of the paper, an optimized iPhone implementation is underway.

The axis of the corridor has lines which are parallel that represent the transition from the walls to the floor and feature lines, such as the door frames, intersect these lines but are perpendicular to them. Thus, the matching problem is reduced to matching the distances between the feature lines. Furthermore, full 3D maps of the corridor can be created and referenced. Since from each single image, the 3D measurements of the corridor (the floors and doors, turns, etc) have been obtained, we will use the location measurement data (such as accelerometer and magnetic components) from a smartphone to roughly align the local models and then by matching the local floor and door models, we refine the localization results and generate a global 3D wireframe model. Information about the device/user can also be interpreted from the data. The differences in capture times and change in frame of the images allows for a determination of speed. Since the position of the camera is easily found relative to the 3D model of the corridor, our method can even be used to provide real-time 3D SLAM in a corridor using only the hardware of a smartphone.

Due to the low processing requirements and power consumption, our method makes 3D reconstruction of a corridor in real-time possible on a basic smartphone. With the increasing ubiquity of such devices, the techniques proposed here open up many useful applications. Even a beginner in amateur robotics would have access to a smartphone and this allows even the most inexpensive of mobile robots a method by which to navigate corridors. A visually impaired person would not need any special equipment to find their way through a building using this method; all they need to

do is take out their phone to get real-time information on their heading. Such a low resource technique would allow such navigation techniques to leave the labs where they're being researched and actually be used in real world applications.

While we have discussed the advantages of our proposed method in cases where there are low resources, the power of the proposed techniques in high resource applications should not be overlooked. Even with more advanced robotics equipment, combining real-time 3D modeling along with high speed mobility is a difficult task. On high power equipment, our technique should produce more than enough 3D models despite any high speeds. Furthermore, 3D range sensors always have distance limitations where our proposed methods have virtually no distance limit. Therefore, our method can also be used, with very little additional computation, to produce preliminary results for information beyond the range of the 3D sensors.

6. Conclusions And Future Work

In this paper, we have presented a real-time, low resource 3D corridor reconstruction method from a single camera. We have also proposed the perspective based Hough transformation algorithm which allows for the fast reconstruction found in this paper. We have shown that by exploiting the characteristics of a corridor, accurate 3D models can be generated quickly with low resources.

In future work, this method can be used in a wide range of applications. It provides a powerful robotics navigation system from nothing more than a smartphone. The visually impaired can be guided through a corridor to find the doors they are seeking simply by using their phone as well.

Additionally, this work has the potential to function in environments with properties analogous to corridors such as highways, sidewalks, city blocks, etc. More generally, the techniques of using existing human-made structural properties to model surroundings can be used in a wide variety of situations.

7. Acknowledgements

The authors want to thank all members of the City College Visual Computing Research Laboratory for their continuing support, advice, and opinions of the research. We thank the anonymous reviewers for their constructive comments and suggestions that help to improve the presentation of the paper. We also thank the support by the National Science Foundation (Award # EFRI-1137172) and PSC/CUNY Cycle 44 Research Program (Award # 66786-00 44).

References

[1] O. Barinova, V. Konushin, A. Yakubenko, K. Lee, H. Lim, and A. Konushin. Fast automatic single-view 3-d reconstruction

of urban scenes. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision ECCV 2008*, volume 5303 of *Lecture Notes in Computer Science*, pages 100–113. Springer Berlin Heidelberg, 2008. 2

[2] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000. 2

[3] E. Delage, H. Lee, and A. Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image, year=2006, month=, volume=2, pages=2418-2428, doi=10.1109/CVPR.2006.23, issn=1063-6919,. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. 2

[4] E. Delage, H. Lee, and A. Ng. Automatic single-image 3d reconstructions of indoor manhattan world scenes. In S. Thrun, R. Brooks, and H. Durrant-Whyte, editors, *Robotics Research*, volume 28 of *Springer Tracts in Advanced Robotics*, pages 305–321. Springer Berlin Heidelberg, 2007. 2

[5] H. Du, P. Henry, X. Ren, M. Cheng, D. B. Goldman, S. M. Seitz, and D. Fox. Interactive 3d modeling of indoor environments with a consumer depth camera. 2

[6] R. Ebrahimpour, R. Rasoolinezhad, Z. Hajiabolhasani, and M. Ebrahimi. Vanishing point detection in corridors: using hough transform and k-means clustering. *Computer Vision, IET*, 6(1):40–51, January 2012. 2

[7] J. Forsberg, U. Larsson, and A. Wernersson. Mobile robot navigation using the range-weighted hough transform. *Robotics Automation Magazine, IEEE*, 2(1):18–26, Mar 1995. 2

[8] M. B. J. Larsson. Purdue experiments in model-based vision for hallway navigation. *IROS, In Proceedings of the Workshop on Vision for Robots*, pages 87–96, 2005. 2

[9] A. Kosaka and J. Pan. Purdue experiments in model-based vision for hallway navigation. In *In Proceedings of the Workshop on Vision for Robots in IROS'95*, pages 87–96, 1995. 2

[10] J. e. a. Lee. Door detection algorithm of mobile robot in hallway using pc-camera. *International Symposium On Automation And Robotics In Construction*, 2004. 2

[11] R.-C. Lo and W.-H. Tsai. Perspective-transformation-invariant generalized hough transform for perspective planar shape detection and matching. *Pattern Recognition*, 30(3):383–396, 1997. 2, 4

[12] M. M. Nevado, J. G. Garca-bermejo, and E. Z. Casanova. Obtaining 3d models of indoor environments with a mobile robot by estimating local surface directions. In *Robotics and Autonomous Systems*, 48:131–143, pages 131–143, 2004. 2

[13] M. Sun, D. Zhang, and Z. Hu. 3d tunnel model reconstruction from monocular image, 2009. 2, 4

[14] Y. Tian, X. Yang, and A. Arditì. Computer vision-based door detection for accessibility of unfamiliar environments to blind persons. In K. Miesenberger, J. Klaus, W. Zagler, and A. Karshmer, editors, *Computers Helping People with Special Needs*, volume 6180 of *Lecture Notes in Computer Science*, pages 263–270. Springer Berlin Heidelberg, 2010. 2