UNIVERSITEIT VAN AMSTERAM

BACHELOR THESIS

A Color Based Rangefinder for an Omnidirectional Camera

Author: Quang Nguyen Supervisor: Arnoud Visser



June 26, 2009

A Color Based Rangefinder for an Omnidirectional Camera

Bachelor Thesis (Afstudeerscriptie)

written by

Quang Nguyen, 5619793 (born Februari 3rd, 1988 in Assen, the Netherlands)

under the supervision of Arnoud Visser

submitted in partial fulfillment of the requirements for the degree of

Bsc Kunstmatige Intelligentie

at the Universiteit van Amsterdam.

Abstract

This paper proposes a method to use the omnidirectional camera as a rangefinder by using color detection. The omnicam rangefinder has been tested in USAR-Sim for its accuracy and for its practical use to build maps of the environment. The results of the test shows that an omnidirectional camera can be used to accurately estimate distances to obstacles and to create maps of unknown environment.

Contents

1	Inti	roduction	1						
2 Related Work									
3	Method								
	3.1	Free space pixel identification	3						
		3.1.1 Collecting training data	3						
		3.1.2 Color Histogram	5						
	3.2	Omnicam Rangefinder	5						
		3.2.1 Polar scanning	5						
		3.2.2 Measuring distance	6						
		3.2.3 Outlier Rejection Filter	8						
4	Exp	perimental Setup	8						
	4.1	USARSim	8						
	4.2	Accuracy Test	9						
	4.3	Map Building Test	9						
		4.3.1 Environments and the Agent	9						
5	Results 1								
	5.1	Accuracy Results	12						
	5.2	Map Building Results	14						
6	Dis	cussion and Future work	18						
7	Cor	nclusion	20						

1 Introduction

An important aspect of robotics is the task of collecting detailed information about unexplored or disaster struck area's. A part of this task is to make a robot measure distances to obstacles in order to localize itself and to create a map of the environment. This paper proposes the use of an omnidirectional camera combined with a color based free space classification system to create a rangefinder which can estimate these distances. 'Can an omnidirectional camera be used effectively as a rangefinder?' is the research question of this paper.

Until now, active sensors like a laser scanner are being used as rangefinders. Active sensors tend to have highly accurate measurements, however they also have a limited range and a limited field of view. Another problem is that they are heavy and can consume lots of energy. Therefore using an active sensor on a small or flying robot is not a viable choice.

This is why it is important to focus attention on alternative sensors like a passive sensor. A passive sensor like the visual sensor do not have the disadvantages of an active sensor. Furthermore a visual sensor can have a 360° field of

view by using an omnidirectional camera. However a visual sensor tends to be inaccurate compared to a laser sensor and it is hard to estimate the depth on an image using an omnidirectional camera.

Again, this paper proposes a method to implement a rangefinder for an omnidirectional camera. This rangefinder uses a color histogram, which is a color based statistical model, to identify free space in the immediate surroundings. Furthermore a comparison between an omnicam and a laser determines which sensor, in terms of accuracy and practical use, performs better in what kind of environment or circumstance. Practical use is tested by letting a rangefinder sensor build a map of an environment. Scanmatching algorithms are used in combination with the rangefinder in order to estimates its position on the map, which is an important part of mapping unknown environments. The implementation and testing of this method is done in USARSim [1], which is a simulation environment that can be used as a research tool [2].

Section 3 describes the theory and method that have been used for the development and validation of the omnicam rangefinder. Section 4 describes how the experiments are set up to test the omnicam rangefinder. Section 5 handles the outcome of the test results while section 6 describes the discussion and further work. The final section makes a conclusion of this research.

2 Related Work

Many methods of using a visual sensor to detect obstacles have already been created. The winner of the DARPA challenge 2005 used a visual sensor as an obstacle detection sensor [3]. Rauskolb et al. [4] improved this obstacle detector so it can also be used in urban environments.

Since the introduction of the omnidirectional camera in USARSim [5] many new applications have been designed for the omnicam. Roebert used the omnidirectional camera to create a bird-eye view map of an environment [6]. However these bird-eye view maps of the environment are not usable by autonomous robots for navigation. The robots need to have an obstacle detector, know what the free space is and have the ability to create a map where the free space and obstacles are shown in order to navigate through an environment.

Maillette de Buy Wenniger [7] created a free space detector that uses a color histogram trained with the colors of free space to identify free space in a bird-eye view image. Scaramuzza [8] has designed a rangefinder for an omnicam sensor that can detect and measure the distance to obstacle points in a black and white environment. Together with localisation performed by scanmatching algorithms, best desribed by Slamet [9], these three components can be combined to create a map of an environment.

3 Method

This section describes the method and theory behind the omnicam rangefinder. The theory behind the rangefinder is to use a trained color histogram to classify free space pixels in an image. By using this knowledge the rangefinder can also find the non free space pixels of an image. Maillette de Buy Wennigers work [7] has already shown that a color histogram can be used for identifying free and non-free space pixels.

This paper suggests an omnicam rangefinder which uses the free space detection system of Maillette de Buy Wenniger to classify pixels as either free space or non-free space. From there on the rangefinder uses this knowledge to detect boundary points of an obstacle by using polar scanning combined with false negative and false positive filters. The robot can then estimate the metric distance between the robot and the boundary point. After the distances have been estimated an outlier rejection filter is used to reject the estimated distances that have a high probability of being inaccurate.

3.1 Free space pixel identification

The color histogram is a statistical model used to identify free space pixels based on their color values. It needs to be trained by using a set of training data which consists of a collection of pixels where the class, which can be free space or non-free space, is already known. A trained color histogram can then calculate the probability if a certain color value belongs to a certain class.

3.1.1 Collecting training data

When the robot gets dropped into an unexplored environment it is going to need a laser scanner to create the training set for the color histogram. After creating the training set the robot can venture further into unexplored area's without using the laser scanner to measure the distances to obstacles. In order to build the training set the robot needs to provide a bird-eye view image of the environment [6] and the laser measured distances image of the environment. These two components are then fused to create a bird-eye view image where the free space is shown. The pixels that are classified as free space on that image are used to train the histogram. This entire procedure is visualized in figure 1. The laser rangefinder has a field of view of 180° compared to the omnicams 360° , therefore the created bird-eye view images of the omnicam has been cut in half to make it synchronize with the laser image.

A bird-eye view image can be created by following Nayar's described relation between a pixel on an omnicam image and the corresponding pixel on the birdeye view image if nothing obstructs the view [10]. A pixel on an omnicam image is described by $p_{omn} = (x_{omn}, y_{omn})$ and a pixel on the bird-eye view image is described by $p_{be} = (x_{be}, y_{be})$. These are the equations:





Figure 1: Procedure to create trainingdata. The half circle on the left part of the images is from the omnicam, this will not be used for the training data

$$\theta = \arccos \frac{z}{\sqrt{x_{be}^2 + y_{be}^2 + z^2}} \tag{1}$$

$$\phi = \arctan \frac{y_{be}}{x_{be}}, \rho = \frac{h}{1 + \cos \theta} \tag{2}$$

$$x_{omn} = \rho \sin \theta \cos \phi, \ y_{omn} = \rho \sin \theta \sin \phi \tag{3}$$

The constant h is the radius of the circle describing the 90° incidence angle on the omnicam effective viewpoint. The variable z is used to describe the distance between the effective viewpoint and the projection plane in pixels. Roeberts work [6] has shown that accurate bird-eye view maps can be achieved by using these formula's.

An image of the laser measured distances can be constructed by using a laser rangefinder to measure the distances on every scanline between a 'hit point' and the origin of the laser, which is the sensor position on the robot. If the distance of the 'hit point' is bigger than the detection range, the laser will return the maximum detection range for that scanline. Multiple laser measurements can be done to increase the probability that free space is detected on a certain scanline. These measured distances can then be converted to an image like in figure 1b.

Finally a combined image between these two components can be created and

the color pixels that are classified by the laser rangefinder as free space on that image can be used as the trainingdata for the color histogram.

3.1.2 Color Histogram

The color histogram trains itself by counting how many times a certain RGB value exists in the trainingdata. This trained color histogram is usable to classify free space pixels in new omnicam images. The trained color histogram uses this discrete probability distribution to classify a pixel as either free space or non free space:

$$P_{HIST}(rgb) = \frac{c[rgb]}{T_c} \tag{4}$$

Here, c[rgb] returns the count of the histogram bin that is associated with the rgb color. T_c returns the total count of all the bins of the histogram. The outcome of a particular color will thus have a value between 0 and 1. In order to filter out the false positives the histogram uses a *Probabilitythreshold* which is set as $0 \le \theta \le 1$. Therefore a pixel is considered as free space if

$$P_{HIST}(rgb) \ge \theta \tag{5}$$

3.2 Omnicam Rangefinder

The basis of this method has been derived from Scaramuzza's black and white omnicam rangefinder [8]. However the proposed omnicam rangefinder in this paper uses the color based free space detection described in section 2.1 instead of only detecting black and white colors. This free space detector is combined with the rangefinders very own detection method which uses polar scanning combined with false positive and false negative filters to detect pixels of an obstacles boundary point. These detected pixels are considered as the hit points of the scanlines. Having detected a hit point the rangefinder estimates the metric distance between the robot and that hit point. At the end of this method an outlier rejection filter is used to reject the estimated measurements that have a high probability of being inaccurate.

3.2.1 Polar scanning

The omnicam rangefinder uses polar scanning to create scanlines that are coming from the center of an omnicam image, a visualisation can be found in figure 2b. Every pixel in each scanline gets classified as either free space or non-free space according to the trained color histogram. The omnicam rangefinder then uses its false positive and false negative filters to find the correct hit point for each scanline. These filters uses a variable parameter which depends on the current environment or situation.



(a) The original image how the omni- (b) The same image with scanlines vicam rangefinder receives it sualized on it

Figure 2: A visualization of scanlines on an omnicam image

- 1. False positive filter: This filter determines if a non-free space pixel is a hit point by checking if N pixels behind it are also classified as non-free space pixels. An example of this filter can be found in figure 3a.
- 2. False negative filter: This makes sure that a candidate hit point does not get rejected because a free space pixel, which was actually misclassified, interrupts the sequence of non-free space pixels described in the false positive filter. The parameter K determines how long the sequence of free space pixels should be before rejecting the candidate hit point. The visualisation is found in figure 3b.

The metric distance from the robot to a pixel is calculated whenever that pixel is classified as a hit point according to the rangefinder. When there is no hit point on a scanline the rangefinder returns the maximum range, which is another variable parameter. When the hit point is too close to the position where the robot is, this would make the measured distance unreliable, the rangefinder returns the minimum range, which is also a variable parameter.

3.2.2 Measuring distance

The rangefinder calculates the metric distances to each hit point that it can find. The metric distance is calculated by using this formula which Scaramuzza [8] has also used for its rangefinder and can also be derived by using basic math knowledge.

$$d(\rho) = h \tan(\frac{\rho}{\alpha}) \tag{6}$$

 ρ represents the pixeldistance from the hit point to the robot. α is a constant value that depends on the mirror shape and the camera-mirror distance. h is the height in meters from the groundfloor to the effective viewpoint of the hyperbolic mirror. The metric distance d is calculated in meters. Sample distance measurements are needed in order to approximate the constant α . This can be



(a) Example where N=4: The pixel with an 'X' has been classified as a hit point because N pixels behind it are classified as non free space pixels



(b) Example where K = 2: Hit point has not been found because there is more free space starting from the 'O' pixel. This is because K pixels behind it are also classified as free space pixels. Note that the free space pixel between the two non-free space pixels is negated because the next pixel behind it is classified as a non-free space pixel

Figure 3: A representation of a set pixels from a scanline, green pixels are classified as free space and red pixels are classified as non-free space pixels

achieved by letting a laser range sensor measure a distance in meters and letting the omnicam sensor measure that same distance in pixels.



Figure 4: The basis of the pixel to meters formula lies in the blue rectangular triangle

This formula had been derived by using figure 4. The blue rectangular triangle is used to derive the following equation:

$$d(\theta) = h \tan(\theta) \tag{7}$$

The relation between ρ , the distance in pixels, and the unknown angle θ can be approximated using a first order Taylor expansion.

$$\theta \approx \frac{1}{\alpha}\rho$$
(8)

The combination of equations 7 and 8 outputs equation 6. Unfortunately, because of the hyperbolic shape of the mirror this formula is limited to mostly small distances. The formula is unusable when $\theta \ge 90^{\circ}$. The metric distance would be then be negative and hitting the hyperbolic mirror with a ray of light at an angle higher than 89° would not reflect the light ray onto the camera image plane.

3.2.3 Outlier Rejection Filter

A robot equipped with an omnicam rangefinder needs to localize itself when it is going to map an unknown environment. This is done by using scanmatching algorithms, these algorithms need to use the measured distances from a rangefinder in order to do their localisation task. Scanmatching algorithms are designed to find hit point correspondences between an old range measurement and a new range measurement. By finding these point correspondences the scanmatching algorithm can estimate its new pose on the map.

Because of their method to estimate new poses these algorithms are very sensitive to outliers. Outliers are most of the time inaccurate measurements and having too many outliers is going to confuse a scanmatching algorithm which of course results into inaccurate localisation.

Using an outlier rejection filter reduces the chance that a scanmatching algorithm will breakdown. The filter works by first calculating the average measured distance, then the minimum and maximum boundaries are set by adding and subtracting N meters from the average distance. The rangefinder then considers the distance measurements that were outside the boundaries as outliers. These outliers are of course rejected because they have a high probability of being an inaccurate measurement.

4 Experimental Setup

This section describes the testing environment and the experiments that are used to test the omnicam rangefinder. The implementation of the omnicam rangefinder and the testing of it were done in USARSim which is a simulation environment. The omnicam rangefinder was tested for its accuracy and for its practical use to create a map of an environment.

4.1 USARSim

USARSim is a 3-D simulation environment that can simulate real world environments and situations. This program is intended as a research tool to study the use of robots in the real world. Robots in USARSim can therefore use simulated realistic tools to complete their tasks. Experiments have shown that perception algorithms that are developed in USARSim can easily be converted and used for the real world [2].

4.2 Accuracy Test

This first experiment compares the measured ranges from an omnicam against the measured ranges from a laser. The laser sensor is highly accurate and can therefore be used as a reference measurement. The ranges are collected by letting the robot drive around in an environment while measuring the distances using both the omnicam and the laser. Each scanline measurement from the laser gets compared to the appropriate scanline measurement of the omnicam. Then by subtracting the omnicam measurements from the laser measurements one can get the differences between them. These differences can then be plotted in a histogram to show the omnicams accuracy compared to the lasers accuracy.

4.3 Map Building Test

The second experiment tests if the omnicam sensor can actually be used for localization and building maps. The sensor has to create accurate maps of environments in order to achieve this. Again, the laser sensor is used to create a reference map of the environment. The omnicam then drives exactly the same route as the laser to create an omnicam map. The omnicam map is then compared with the reference map to determine how different the map is.

The mapbuilding is done using two different ways:

- 1. Using Deadreckoning with the GroundTruth as the seed. This setting makes sure that the robot always knows where it is on the map. This means that the robot can always localize itself without scanmatching, therefore the quality of the map fully depends on the accuracy of the sensor.
- 2. Using Quad Weighted ScanMatching (QWSM) [11] with an Inertial Navigation System (INS) as the seed. INS uses the robots motion sensors to calculate its new pose compared to the old one. Therefore the robot depends on INS and scanmatching for estimating its current location. This setting is a more realistic setting and must be used when the robot is venturing into unknown terrain.

Using QWSM and INS results in a less accurate map, but the omnicam needs to be evaluated with this setting in order to proof that it can be used in unknown terrain.

4.3.1 Environments and the Agent

Two different environments are used to test the omnicam rangefinder. The first one is a grassmaze as shown in figure 5a. This environment consists of very



Figure 5: These are the environments in USARSim to train and test the omnicam rangefinder.



Figure 6: These are some of the objects in the factory environment

small corridors, lots of turns and the hedge has about the same color as the floor, which is made out of grass. The challenge is to correctly map the small corridors and to make a distinction between the hedge and the grass.

The second environment is a factory as seen in figure 5b, this environment has big corridors, less turns and it contains a lot of different objects. The challenge in this map is to clearly detect the objects, figure 6 shows a couple of examples of these obstacles.

The name of the robot that is used to do these experiments is the OmniP2DX(Figure 7), this robot comes equipped with a laser and an omnicam sensor. The height of the omnicams effective viewpoint to the ground is 0.931m, this value is needed to calculate metric distances.

Before starting the experiments it is important to estimate what the maximum range of the OmniP2DX is. This can be done by estimating the value of the constant α in equation 6. The α is estimated by rewriting the equation and performing several sample distance measurements by letting a range sensor measure a distance in meters and letting the omnican sensor measure that same distance in pixels. Having estimated the α the distance formula can be used to plot the relation between the pixel distance and the metric distance. Figure 8



Figure 7: OmniP2DX, the omnicansensor is mounted on top of the pillar

shows this relation and it also shows that the metric distance difference between pixels exponentially increases. This means that having an off-by-one-error misclassification on a distance far away from the robot can result into a high metric distance error. Setting the maximum range at 3.8 meters is therefore a good tradeoff between the maximum range and the accuracy of the omnicam.



Figure 8: The relation between the pixel distance and the metric distance

5 Results

This section deals with the results of the experiments with the omnicam rangefinder. The results have been obtained by using the parameters stated in table 1. These are the optimal parameters that have been derived by hand. At the time of

Map	ScanLines	Nonfreepixels	Groupedpixels	Probabilitythreshold	Rejectiondistance
Grassmaze	360	20	2	0.05	1.25m
Factory	360	20	2	0.075	1.25m

Table 1: Measurementoutput statistics

writing there exists no learning algorithm for finding the optimal parameters for every situation. The laser sensor used for these experiments is the SICK laser which has a maximum ange of 19.8 meters compared to the omnicams maximum ange of 3.8 meters. The image resolution for the omnicam images used for this project is 1024x768.

5.1 Accuracy Results

Figures 9a and 9b shows the results of the accuracy test of the omnicam rangefinder for respectively the grassmaze and the factory. Because of the skewed data of the grassmaze the histograms needed to be shortened by discarding all the measurement errors higher than 1 meter. Table 2 shows how many measurements were actually discarded from the histogram. The table shows that the omnicam sensor makes bigger measurement errors in the grassmaze. These errors can get up to 2.5 meters. Table 3 shows the values of how much skewed the data really is. The factory measurements are quite stable whereas the grassmaze is skewed, this can also be seen in the histograms of figure 9.

Figure 10 shows the relation between the accuracy of the omnicam range finder and the reference metric distance that it needs to estimate. The figure shows that the omnicam range finder is measuring longer distances much further than they in fact are. There is only a 3D histogram of the grassmaze because the factory did not have enough long distance measurements to create a 3D histogram where this relation can be seen.

The measurements of the factory environment were done in the big and long corridors, this means that the factory's histogram is showing the accuracy of the omnicam in a more optimal situation. The environment did not have a lot of corners compared to the grassmaze environment which was chosen to stresstest the omnicam rangefinder.

Table 4 shows that the omnicam rangefinder has an average absolute accuracy difference of 8.09cm in the factory compared to 13.75cm in the grassmaze. The table also shows how many distance measurements were used in order to create the histograms. The distance measurements in the grassmaze are considerably more than in the factory. The route driven with the robot in the factory was therefore shorter. Again, the factory route was chosen to show the results in an optimal situation while the grassmaze route was chosen to stresstest the omnicam rangefinder.



(b) Factory: Histogram of the omnicam accuracy

Figure 9: The omnicam accuracy compared to the laser accuracy measured in both environments



Figure 10: Grassmaze: The figure shows the relation between the omnicam accuracy and the reference distance of the laser sensor. The omnicam tends to get less accurate when measuring longer distances.

Map	No. of Differences $\leq -1m$	No. of Differences $\geq 1m$	No. of Differences $\geq 2.5m$
Grassmaze	443	131	21 %
Factory	3	1	0 %

Table 2: Table of left out differences

5.2 Map Building Results

Factory environment

Figure 11a and 11b shows the results of building a map of the factory environment using an omnicam sensor and a laser sensor combined with deadreckoning and the groundtruth for localisation. Because of its accuracy the laser created map serves as an indication for what the groundtruth map should look like. Comparing both maps shows that the omnicam map does not differ that much from the laser created map. The black dots and lines on the map represents detected obstacles, the gray color represents the cleared space while the white color represents the free space detect by the rangefinder. Having free space on the map does not mean that there are no obstacles at that area. It serves as an indication of where the robot can explore and by doing this the area changes into cleared space.

A remarkable difference between both maps is within the red rectangle. The omnicam map has a wall drawn at that location but the laser does not have it drawn. However the omnicam map is correct at this situation, there is indeed a

Map	Skewness of differences data	Skewness of the laser measured data
Grassmaze	2.3750	1.0925~%
Factory	-0.3747	0.6555~%

Table 3: Table with the skewness values

Мар	No. Measurements	Avg Absolute Difference	Avg Percentage Difference
Grassmaze	31324	0.1375 m	7.639%
Factory	9334	0.0809m	4.493%

Table 4: Table with Measurement statistics

big object(figure 6a) which the laser looked right through because the cabinet was empty. The lava pipe from figure 6b, which is represented by the most upperleft wall on the map, was detected by both sensors.

Figure 12 shows the results of the same route using QWSM scanmatching with INS. Comparing these maps with the maps from figure 11 shows that the map from figure 12b is more accurate than the map from figure 12a. These maps show that localisation with the omnicam sensor performs worse than localisation with a laser sensor for this situation.

A reason for this is that the corridors are too wide, the omnicam can not detect the walls on the other side of the corridor because of its limited range of 3.8 meters compared to the lasers range of 19.8 meters. By not detecting these walls the localisation goes wrong because the scanmatching algorithms has a harder time to detect what has changed in the environment.

Another reason is that the scanmatching algorithms that were used were tweaked for the laser range sensor.

Grassmaze environment

Figure 13 shows the results using deadreckoning with the groundtruth to build a map of the grassmaze. Because of the turns in the grassmaze the SICK laser loses its advantage to measure distances up to 20 meters. The robot started in the middle of the maze and made its way to the exit of it. Apart from the noise on the path both maps look quite similar to each other. The laser could sometimes look through the hedge because of the very small holes in the hedge. The laser would then see the path on the other side but it can also create some noise like the noise in the upper part of its map. The omnicam map does not have those problems.

Figure 14 shows the created maps of the grassmaze with QWSM and INS. The localisation for the laser sensor went wrong on this map. This is because the laser measurements were too unstable, mostly because it looked right through hedges which creates outliers that can breakdown the scanmatcher. The omnicam map is reliable, the map only has a small curve which can be seen when it is compared with the maps from figure 13. The omnicam could localise itself



Figure 11: Factory map created with deadreckoning and groundtruth



(a) Omnicam sensor factory map

(b) Laser sensor factory map

Figure 12: Factory map created with QWSM and INS



(a) Omnicam sensor grassmaze map



(b) Laser sensor grassmaze map

Figure 13: Grassmaze map created with deadreckoning and groundtruth

within the grassmaze because it did not had many outliers thanks to the outlier rejection filter and it did not look through hedges like the laser sensor because the holes of the hedges were too small to be even visible on the omnicam.

The grassmaze proofs that the omnicam combined with localisation can work in an environment with narrow corridors, lots of corners and a wall which looks like the groundfloor.

6 Discussion and Future work

Accuracy

The difference in accuracy between an omnicam and a laser lies in the detection of object boundaries. Using color distinction to detect these boundaries does not provide perfect results. Using color detection can sometimes get a pixel misclassified as a hit point. These misclassifications are fatal for long distance measurements because of the hyperbolic shape of the mirror of the omnicam. A misclassified pixel can therefore result in a measurement error to up to 2.5 meters, this can also be seen in table 2.

The accuracy tests also shows that the omnicam measures longer distances further away than they in fact are. This is a good result showing that the omnicam rangefinder can create reliable maps. This is because the inaccurate measurements can better be drawn outside of the path instead of on the path, which makes the map noisier. Having error measurements on shorter distances does not matter much because figure 8 shows that the metric distance error between those pixels is very small.

Future work for the accuracy improvement using color detection is trying other color spaces. This is also useful to let the omnicam rangefinder work in other area's where the RGB color space might fail. A fast way to increase accuracy is to use a higher image resolution, however higher resolution images need more time to process.

Corner misclassification

The maps created by the omnicam shows that there exists misclassification at the beginning and at the end of each corner where a turn can be made. The omnicam misclassifies pixels when it is approaching a turn. The reason for this is that the omnicam rangefinder does not know if there is a corner by just looking at one image. It could just as well be the wall, therefore when it approaches a corner it already draws obstaclepoints on the map which are in fact misclassifications. This undesirable effect of the omnicam sensor does not pose a threat when the omnicam has to navigate through wide corridors. However having noise in narrow corridors can certainly result in an inaccurate map.

This problem might be solveable by implementing a method that can detect corners on a sequence of images. Another method is to create a function that



(a) Omnicam sensor grassmaze map



(b) Laser sensor grassmaze map

Figure 14: Grassmaze map created with QWSM and INS

can classify and erase noise on the map. This might be achievable because the corner noise consists of only a few dots as shown on the omnicam maps. A real object or wall is represented by very thick lines on the map.

Thick lines

This next problem of the omnicam is a result of its inaccuracy. The omnicam draws thick lines because the distance measurements taken from different angles are not relative to each other. This leads into multiple obstaclepoints on the map that are close to each other instead of points that overlap with each other like with the accurate laser sensor. These inaccuracies therefore results into a thick line on the map. This problem needs to be solved because navigating through narrow corridors is undesirable if the thick lines are drawn half on the pathway.

The solution for this problem is creating a function that can smooth thick lines. The function would first need to identify the thick lines on the map so it can turn those thick lines into thin lines by taking the average of its points.

Automatic parameter learning

The experiments were set up by using parameters that were derived by hand. Future work would therefore include an automatic parameter learning algorithm. An approach would be to use the trainingset of the histogram to train the rangefinder parameters.

Further Improvements

The omnicam rangefinder does not work on a slope or other height differences. This is because the pixel to meters formula does not incorporate height differences, this can be seen in figure 4. Further improvements lies in the theory of detecting free space based on color detection. The free space detection does not work well when the walls or objects all have the same color. This situation might happen literally but having a dark room can also create this situation. The free space detector might also break down when it has to deal with different lighting conditions, a solution for this problem might be to adapt the color histogram to a HSV color space because this color space can remove color intensity. Future work should also include a method that can detect when the floor color changes.

7 Conclusion

Based on the results found in section 5 it can be concluded that an omnidirectional camera can be used efficiently as a rangefinder. The omnicam rangefinder can have an inaccuracy of 8.09cm compared to the laser sensor. The results also shows that the omnicam rangefinder can use scanmatching algorithms that were tweaked for a laser sensor to create accurate maps of an environment eventhough it does nog possess the precise accuracy of a laser sensor. The omnidirectional camera can also detect obstacles, like an empty cabinet(figure 6a), that a laser sensor can not detect. The omnicam rangefinder is based on color detection and is therefore unusable when the obstacles have exactly the same color as the floor.

This research can lead to the development of small and flying robots that can use the lightweight, energy efficient and inexpensive omnidirectional camera to quickly create a map of an environment. These robots need to work together with a robot mounted with a laser sensor because the color histogram learned with the laser sensor can be distributed wirelessly to the small and flying robots.

Acknowledgements

At first I would like to thank Arnoud Visser, who was the supervisor of this project, for his help, feedback and ideas for the many problems I had with this project.

I also want to thank Gideon Maillette de Buy Wenniger for helping me with the free space detector and therefore giving me the basis to create the omnicam rangefinder. I would also want to thank Tijn Schmits for helping me with finding the specifications of the OmniP2DX in USARSim.

References

- J.W.S.B. S. Carpin, M. Lewis and C. Scrapper, Usarsim: a robot simulator for research and education, 2007, Proceedings of the 2007 IEEE Conference on Robotics and Automation.
- [2] J.W.S.B. Stefano Carpin, Michael Lewis and C. Scrapper, Bridging the gap between simulation and reality in urban search and rescue., 2006, In G. Lakemeyer, E. Sklar, D.G. Sorrenti, and T. Takahashi, editors, RoboCup 2006: Robot Soccer World Cup X, volume 4434 of Lecture Notes on Artificial Intelligence, pages 112, Berlin Heidelberg New York, October 2007. Springer.
- [3] S. Thrun et al., The robot that won the darpa grand challenge., 2006, Journal of Field Robotics, 23(9):p661 692.
- [4] F.W. Rauskolb et al., An autonomously driving vehicle for urban environments., 2008, Journal of Field Robotics, 25(9):p674 724.
- [5] T. Schmits and A. Visser, Proceedings CD of the 12th RoboCup International Symposium, Suzhou, China, july 2008, To be published in the Lecture Notes on Artificial Intelligence series.
- [6] S. Roebert, Creating a bird-eye view map, Bachelor Thesis, 2008.
- [7] G. Maillette de Buy Wenniger, Identifying free space in a robot bird-eye view, Universiteit van Amsterdam, 2008.

- [8] D. Scaramuzza, Exercise 3: How to build a range finder using an omnidirectional camera, 2007, Autonomous Systems Lab Swiss Federal Institute of Technology, Zurich.
- [9] B. Slamet and M. Pfingsthorn, Manifoldslam: a multi-agent simultaneous localization and mapping system for the robocup rescue virtual robots competition, 2006, Universiteit van Amsterdam.
- [10] S.K. Nayar, CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), p. 482, Washington, DC, USA, 1997, IEEE Computer Society.
- [11] A. Visser, B.A. Slamet and M. Pfingsthorn, Robust weighted scan matching with quadtrees, 2008.