A Color Based Rangefinder for an Omnidirectional Camera

Quang Nguyen and Arnoud Visser

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 USARSim 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.**

I. INTRODUCTION

An important aspect of robotics is the task of collecting detailed information about unexplored or disaster struck areas. 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 laser scanners or sonar are being used as rangefinders. Active sensors can generate highly accurate measurements, but have their limitations. They typically scan a surface, missing obstacles just above or below this surface. Some highly reflective or absorbent surfaces are invisible for these sensors. Further they have limits on their range and field of view. Because it are active sensors, they are relatively heavy and can consume a substantial amount of the robots battery capacity. An active sensor on a small or flying robot is therefore not a viable choice.

This is why it is important to focus attention on alternative sensors like a passive sensor as proposed in this paper. A passive sensor based on omnidirectional camera can have a 360^o 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 estimate 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].

Intelligent Systems Laboratory Amsterdam, Universiteit van Amsterdam, Science Park 105, 1098 XH Amsterdam, The Netherlands

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.

II. RELATED WORK

Several methods of using a visual sensor to detect free space have already been created. The winner of the DARPA challenge 2005 used a visual classifier based on color information to estimate the road ahead [3]. Rauskolb et al. [4] improved this algorithm 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 probabilistic methods to learn the appearance of free space in a bird-eye view image based on the color signature. This free space detector is extensively tested, but not used for navigation or mapping yet. Scaramuzza [8] has designed a rangefinder for a low-cost omnicam sensor that can detect and measure the distance to obstacle points in a simple black and white world. In this paper both approaches are combined with a scanmatching algorithm [9], which allows localization and mapping of an environment purely on visual information.

III. METHOD

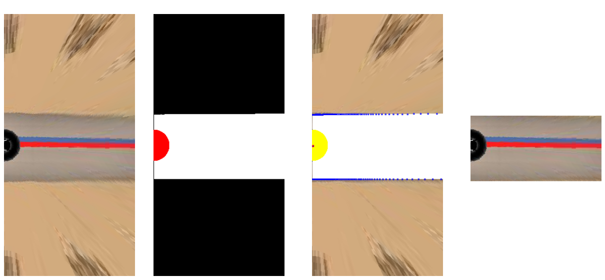
This section describes the method and theory behind the omnicam rangefinder. The theory behind the rangefinder is to use a trained color histogram to classify free space pixels in an image. Maillette de Buy Wenniger's work [7] has shown that a color histogram can be used for reliable identifion of free-space for a simulated and a real robot.

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.

A. 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.

1) Collecting training data: When the robot is deployed 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 areas 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^o compared to the omnicam's 360^o, 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 in the factory environment. (b) Image create by the laser range measurements. (c) The combined image of the bird-eye view image and the laser image. (d) The resulting image that is used for training.

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

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 bird- eye 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:

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

$$\phi = \arctan \frac{y_{be}}{x_{be}}, \quad r = \frac{R}{1 + \cos \theta} \tag{2}$$

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

The constant R is the radius of the circle describing the 90^o 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. Roebert's work [6] has shown that accurate bird-eye view maps can be achieved by using these formulas.

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 training data for the color histogram.

2) Color Histogram: The color histogram is trained by counting how many times a certain RGB value exists in the training data. Before the training can start a decision about the number of bins (n) of the histogram has to be made. In this study histograms with 13 bins are used. A 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 probability threshold is set as $0 \leq \theta \leq 1$. Therefore a pixel is considered as free space if

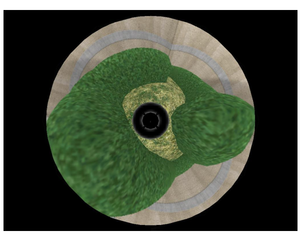
$$P_{HIST}(rgb) \geq \Theta \tag{5}$$

In our experiments, a different threshold Θ was used to accommodate for the circumstances in the different environments (as indicated in table I).

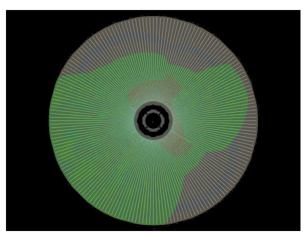
B. Omnicam Rangefinder

The basis of this method has been derived from Scaramuzza's black and white omnicam rangefinder [8]. However the proposed omnicam rangefinder in this article 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.

1) Polar scanning: The omnicam rangefinder uses polar scanning to create scanlines that are coming from the center of an omnicam image, a visualization 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 use a small number of parameters which can be optimized on the environment or situation, as indicated in table I.



(a) The original image of the omnicam.



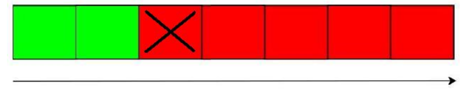
(b) The same image with scanlines visualized on it.

Fig. 2: A visualization of polar scanlines on an omnicam image.

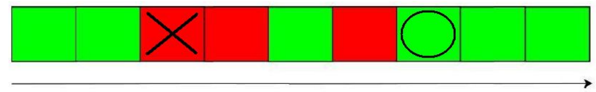
False-positive filter:
: This filter determines if a non-free space pixel is a hit point by checking if N pixels behind it are also classified as non-free space pixels. An example of this filter can be found in figure 3a.

False-negative filter:
: This filter 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. An example is given 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.



(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.

Fig. 3: Illustration of the false-positive and false-negative filters. The figures represent a set of pixels from a scanline, green pixels are classified as free space and red pixels are classified as non-free space pixels.

2) Measuring distance: The rangefinder calculates the metric distances to each hit point that it can find. The metric distance d is calculated by using the formula from Scaramuzza [8]:

$$d = h \tan(\theta) \tag{6}$$

where θ is the incidence angle on the mirror and h is the height in meters from the ground to the effective viewpoint of the hyperbolic mirror. The metric distance d is calculated in meters. The incidence angle θ can be estimated by a first order Taylor expansion $\theta \approx \frac{\rho}{\alpha}$ when the shape of the mirror is not well known. ρ represents the pixel distance from the hit point to the center of the image, α is a constant value that depends on the mirror shape and the camera-mirror distance. The constant α can be estimated by calibrating the omnicam sensor measurements against a laser range sensor measurements. The variables θ, ρ, α and h are illustrated in figure 4. In our case, the shape of the mirror is well known; $\theta = 2\arctan(\frac{r}{R})$.

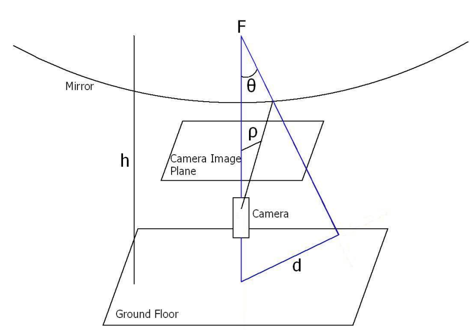


Fig. 4: Illustration of the location of the mirror, image plane and ground floor, as used in equation (6).

Pixels which have a difference larger than 1.25m between the distance estimate of the laser range scanner and the omnicam range scanner were not used for training or evaluation.

IV. 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.

A. USARSim

USARSim is a 3D 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].

B. 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[1] 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. 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 omnicam's accuracy compared to the laser's accuracy.

C. 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 map building is done using two different ways:

- Using Deadreckoning with the GroundTruth as the sensor. This reference 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.
- Using Quad Weighted ScanMatching (QWSM) [11] with an Inertial Navigation System (INS) as the initial pose estimate. INS uses the robots acceleration sensors to estimate the current pose. That pose estimate is checked on consistency with the observations of the range scanner, which results in a new estimate of the current pose. This setting is a more realistic setting and must be used when the robot is venturing into unknown terrain.

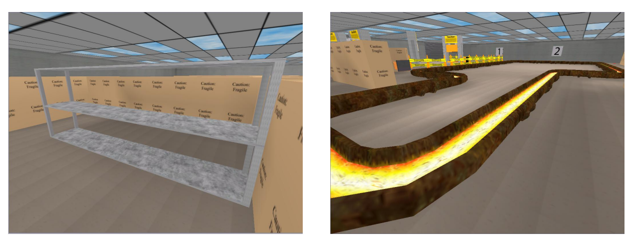
Using QWSM and INS results in a less accurate map, but the omnicam needs to be evaluated with this setting in order to proof that it can be used in a realistic configuration.

D. Environments and robot model

Two different environments are used to test the omnicam rangefinder. The first one is a maze as shown in figure 5a. This environment consists of very small corridors, lots of turns and the hedge has nearly the same color as the floor, which is covered with 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 relatively wide corridors, less turns and it contains a number of unique objects. The challenge in this map is also to detect those unique objects as obstacles, figure 6 shows a couple of examples of these obstacles.

(a) The maze environment. (b) The factory environment.

Fig. 5: Environments in USARSim to train and test the omnicam rangefinder.



(a) A cabinet. (b) A pipe with liquid metal.

Fig. 6: Some unique objects in the factory environment.

The type of robot that is used to do these experiments is the OmniP2DX (Figure 7), which is a robot equipped with a laser scanner and an omnicam sensor. The height of the omnicam's effective viewpoint h to the ground is 0.931m.

Before starting the experiments it is important to choose a maximum range of the algorithm described in section III-B. The maximum range is dependent on the curvature of the mirror of the OmniP2DX. The maximum range can be estimated by rewriting the equation (6) and performing several sample distance measurements by letting a range sensor

[1]A SICK LMS 200 has indoors a statistical error $< 5mm$

Fig. 7: OmniP2DX, with the omnicam sensor high above the Sick laser range scanner.

measure a distance in meters and letting the omnicam 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 shows this relation and it also shows that the discretization error between pixels increases hyperbolically. This means that having an off-by-one-pixel-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. The number of scanlines used was 360.

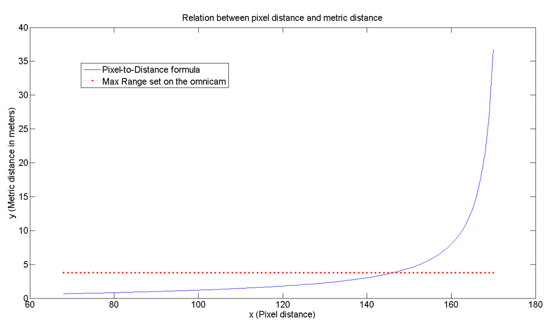


Fig. 8: The relation between the pixel distance and the metric distance.

V. 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 I. These are the optimal parameters that have been derived by hand. At the time of writing there exists no learning algorithm for finding the optimal parameters for every situation is applied. The laser sensor used for these experiments is the SICK laser which has a maximum range of 19.8 meters compared to the omnicam's maximum range of 3.8 meters. The image resolution for the omnicam images used for these experiments is 1024x768 (R=192 pixels).

A. Accuracy Results

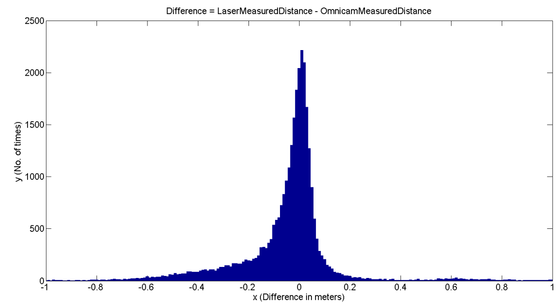
Figures 9a and 9b shows the results of the accuracy test of the omnicam rangefinder for respectively the maze and the factory. The figures show a bigger measurement error is made for the maze, but also that this error mainly due to the tail of the distribution. The factory measurements are quite symmetric whereas the maze measurement are skewed to the left, as can be seen in the histograms of figure 9. For the maze measurements the systematic error is nearly zero, while for the factory measurements the mean of the distribution is a few centimeters to the right.

The measurements of the factory environment were done in wide corridors, which 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 maze environment which was chosen to stresstest the omnicam rangefinder.

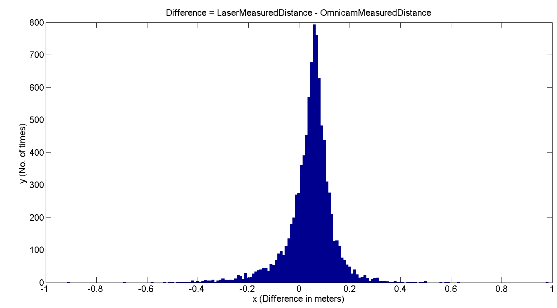
Table II shows that the omnicam rangefinder has an average absolute accuracy difference of 8.09cm in the factory compared to 13.75cm in the maze. Notice that the route driven with the robot in the factory was shorter then the route through the maze, but nicely closes a loop.

Map	Nonfreepixels	Groupedpixels	Probabilitythreshold
Maze	20	2	0.05
Factory	20	2	0.075

TABLE I: Parameters used in the algorithm



(a) Maze: Histogram of the omnicam accuracy

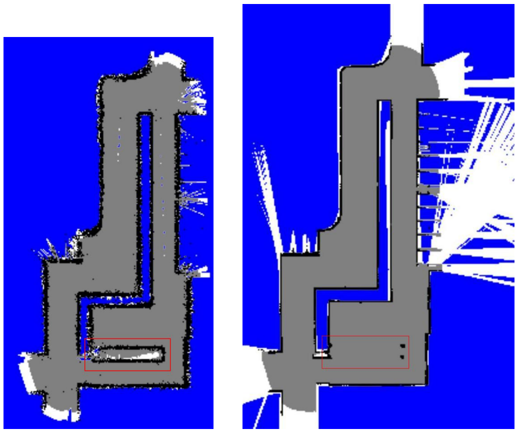


(b) Factory: Histogram of the omnicam accuracy

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

Map	Avg Absolute Difference	Avg Percentage Difference
Maze	0.1375m	7.639%
Factory	0.0809m	4.493%

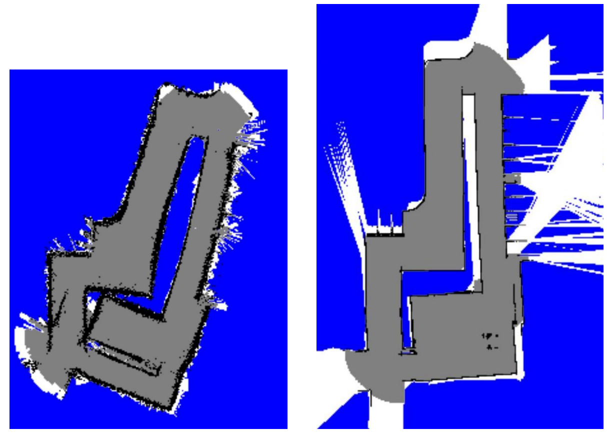
TABLE II: Table with differences between laser and omnicam range measurements



(a) Omnicam sensor factory map

(b) Laser sensor factory map

Fig. 10: Factory map created with localization on ground truth



(a) Omnicam sensor factory map

(b) Laser sensor factory map

Fig. 11: Factory map created with QWSM and INS

B. Map Building Results

Factory environment: Figure 10a and 10b shows the results of building a map of the factory environment using an omnicam sensor and a laser sensor combined with the ground truth (available in simulation) as localization. Because of its accuracy the laser created map serves as an indication for what the ground truth 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 safe space while the white color represents the free space detect by the rangefinder. Both gray and white indicates areas free of obstacles, but grey indicates areas that are well explored, while white indicates areas that could be further explored. The main difference by the maps generated with the omnicam and the laser, is the thickness of the walls. The omnicam map is not as razorsharp as the map generated with the laser scanner.

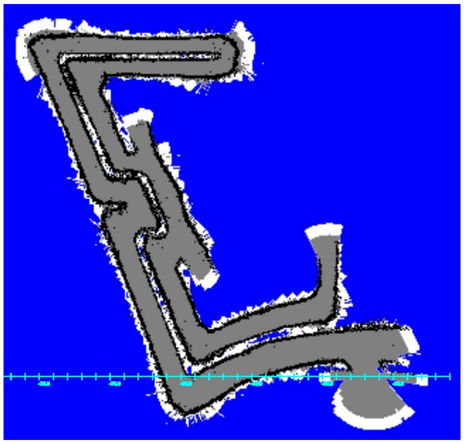
A less obvious difference between both maps is visible at the bottom of the map, indicated with a red rectangle. The omnicam map has found a obstacle at that location while on the laser map only four small dots are visible. The omnicam map is correct at this situation, there is indeed a big obstacle (figure 6a) present on this location. The laser scammer looked right through the cabinet, because no shelf was present at measurement height of the sensor. Another obstacle which had the potential to be difficult to detect, the pipe with liquid metal from figure 6b, was detected without problems by both sensors. This pipe is visible on both maps as the curved upper left wall.

Figure 11 shows the results of the same route using QWSM scanmatching with INS. Comparing these maps with the maps from figure 10 shows that the map from figure 11b is more accurate than the map from figure 11a. These maps show that localization with the omnicam sensor performs worse than localization with a laser sensor for this situation. The reason for this is that if the corridors are relatively wide, the omnicam with its limited range of 3.8 meters sees not that many features. The walls on other side of the crossing are out of range. When the robot is turning on a a location with sparse features, an error in the precise value of this rotation is easily made. In figure 11a small rotation errors are made on multiple locations.

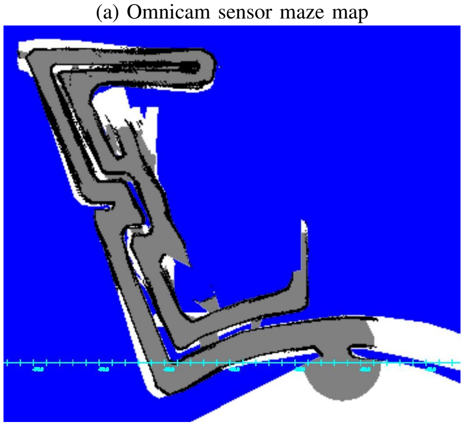
This rotational error could be corrected by post-processing, when loop closure is detected. Note that this are initial results, the used scanmatching algorithms were developed and optimized for the laser range sensor measurements. Although the applied scanmatching algorithms do not have that many parameters, no sensitivity study is performed to study the optimal parameter values for the omnicam range measurements.

Maze environment: Figure 12 shows the results using localization on the ground truth to build a map of the maze. Because of the turns in the maze 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 have a preview on the path on the other side of the hedge. This could be a benefit, but it can also create some distortions like the area in the upper part of its map. The omnicam map does not have those problems.

Figure 13 shows the created maps of the maze with QWSM and INS. The localization for the laser sensor went wrong on this map. This can have several reasons. Firstly, the measurements right through hedges which creates outliers that can breakdown the scanmatcher. Further, the corridors on the outer edge are quite long. The laser range measurements could not detect the end of the corridor, which makes it difficult to notice progress in movement through such corridor. The omnicam has an even shorter range and can also not see the end of the corridor, but detects more structure in the walls which can be used to detect progress in movement.

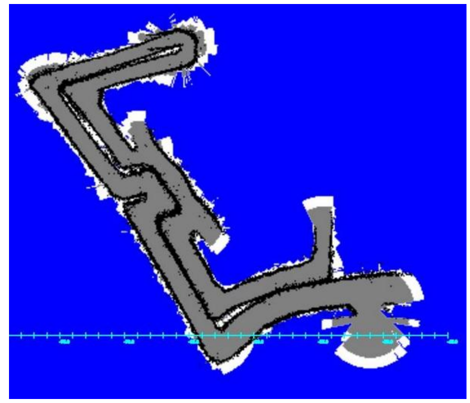


(a) Omnicam sensor maze map

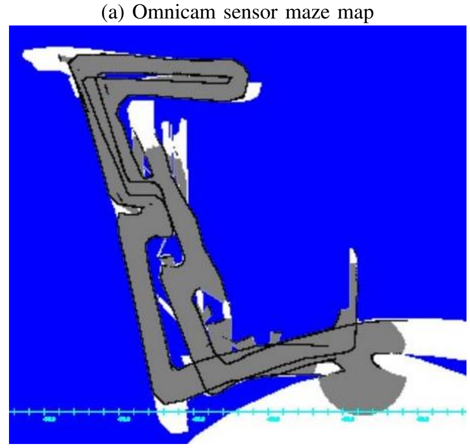


(b) Laser sensor maze map

Fig. 12: Maze map created localization on ground truth



(a) Omnicam sensor maze map



(b) Laser sensor maze map

Fig. 13: Maze map created with QWSM and INS

The omnicam map looks good, the map only has a small rotational error when compared with the maps from figure 12.

The maze proofs that the omnicam combined with localization can work in an environment with narrow corridors, lots of turns and a wall with nearly the same color as the floor.

VI. 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, as indicated in figure 8.

The accuracy tests also shows that the omnicam measures longer distances slightly further away than they in fact are. This is a good way to estimate the location of obstacles, as shown in the maps created by omnicam rangefinder. The first measurements on a large distance are initially drawn a bit away from the free space that could become a path for the robot. When the robot gets near that location, more accurate measurements indicate the precise boundaries of the path. Figure 8 shows that the metric distance error between those pixels on such a small distance 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 areas 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.

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 obstacle points 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 training set 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.

VII. 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 accuracy of 8.09cm compared to the laser sensor. The results also shows that the omnicam rangefinder can use scanmatching algorithms that were optimized for a laser sensor to create accurate maps of an environment even though it does not 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 large robot equipped with a laser sensor to provide a color histogram. This color histogram can be learned by the large robot and distributed wirelessly to the small and flying robots.

Acknowledgements

We want to thank Gideon Maillette de Buy Wenniger for providing inside information about the free-space detection algorithm. We would also want to thank Tijn Schmits for his explanation about the specification and precise construction of the OmniP2DX in USARSim.

REFERENCES

[1] S. Carpin, C. Lewis, J. Wang, S. Balakirsky, and C. Scrapper, "Usarsim: a robot simulator for research and education," pp. 1400–1405, April 2007, proceedings of the 2007 IEEE Conference on Robotics and Automation.

[2] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper, "Bridging the gap between simulation and reality in urban search and rescue." in *RoboCup 2006: Robot Soccer World Cup X*, ser. Lecture Notes on Artificial Intelligence, G. Lakemeyer, E. Sklar, D. Sorrenti, and T. Takahashi, Eds., vol. 4434. Springer, October 2007, pp. 1–12.

[3] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "The robot that won the darpa grand challenge." *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[4] F. W. Rauskolb, K. Berger, C. Lipski, M. Magnor, K. Cornelsen, J. Effertz, T. Form, F. Graefe, S. Ohl, W. Schumacher, Peter, T. Nothdurft, M. Doering, K. Homeier, J. Morgenroth, L. Wolf, C. Basarke, T. Berger, F. Klose, and B. Rumpe, "An autonomously driving vehicle for urban environments." *Journal of Field Robotics*, vol. 25, no. 9, pp. 674–724, 2008.

[5] T. Schmits and A. Visser, "An omnidirectional camera simulation for the usarsim world," in *RoboCup 2008: Robot Soccer World Cup XII*, ser. Lecture Notes on Artificial Intelligence series, vol. 5339. Berlin Heidelberg New York: Springer, June 2009, pp. 296–307.

[6] S. Roebert, T. Schmits, and A. Visser, "Creating a bird-eye view map using an omnidirectional camera," in *Proceedings of the 20th Belgian-Netherlands Conference on Artificial Intell igence (BNAIC 2008)*, A. Nijholt, M. Pantic, M. Poel, and H. Hondorp, Eds., October 2008, pp. 233–240.

[7] G. Maillette de Buy Wenniger and A. Visser, "Identifying free space in a robot bird-eye view," in *Proceedings of the 4th European Conference on Mobile Robots (ECMR 2009)*, September 2009.

[8] D. Scaramuzza and S. Gächter, "Exercise 3: How to build a range finder using an omnidirectional camera," Autonomous Systems Lab Swiss Federal Institute of Technology, Zurich, April 2009, version 1.4.

[9] M. Pfingsthorn, B. Slamet, and A. Visser, "A scalable hybrid multi-robot slam method for highly detailed maps," in *RoboCup 2007: Robot Soccer World Cup XI*, ser. Lecture Notes on Artificial Intelligence, vol. 5001. Springer, July 2009, pp. 457–464.

[10] S. K. Nayar, "Catadioptric omnidirectional camera," in *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. Washington, DC, USA: IEEE Computer Society, 1997, p. 482.

[11] A. Visser, B. A. Slamet, and M. Pfingsthorn, "Robust weighted scan matching with quadtrees," in *Proceedings of the 5th International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster (SRMED 2009)*, July 2009.