University of Osnabrück
Institute of Computer Science
Knowledge Based Systems Research Group

UNIVERSITÄT OSNABRÜCK

University of Hannover
Institute for Systems Engineering
Real Time Systems Group

RTS
Echtzeitsysteme
Universität Hannover
www.rts.uni-hannover.de

Robotic 3D Environment Cognition

Workshop at the International Conference Spatial Cognition

September 2006

Andreas Nüchter, University of Osnabrück
Oliver Wulf, University of Hannover
Kai Lingemann, University of Osnabrück

Contents

Preface

As we set up the workshop with the following words defining the scope

> A fundamental problem in the design of autonomous mobile cognitive systems is the perception of the environment. Robotics researches this field in order to build reliable technical systems or to broaden the understanding of human perception. Perception is therefore studied independently by many researchers. On one hand, a basic part of the perception is to learn, detect and recognize objects, which has to be done with the limited resources of a mobile robot. The performance of a mobile system crucially depends on the accuracy, duration and reliability of its perceptions and the involved interpretation process. On the other hand, automatic environment sensing and modeling is a fundamental scientific issue in robotics, since the availability of maps is essential for many robot tasks.
>
> A revolutionary method for gaging surroundings are 3D laser range finders and 3D cameras, which enable robots to quickly scan objects in a non-contact way in three dimensions. These emerging technologies have lead to new challenges and new potentials for data analysis. Firstly, robotic volumetric or 3D mapping of environments, considering all six degree of freedom of a mobile robot, has been done. Secondly, robots are able to perceive the geometry for avoiding collision in 3D and to identify and stay on navigable surfaces. In addition, 3D sensors have lead to new methods in object detection, object localization and identification.

we were surprised when receiving mostly papers about simultaneous localization and mapping (SLAM). SLAM in well-defined, planar indoor environments is considered solved, but new challenges arise when considering new 3D sensors and more degrees of freedom of representing robot states or environment features. Taking this step into the third dimension, it became evident that current robotic research is coping with these fundamental issues, rather than focusing on high-level recognition and scene understanding.

The following collection covers a wide range of topics, ranging from robotic 3D environment sensing, robot navigation to SLAM. We wish the reader interesting ideas and are looking forward to continuing our research in 3D robotic environment cognition.

As the organizers of the workshop we would like to thank Christoph Hölscher, University of Freiburg, for the support he has given in the last weeks. Furthermore we would like to thank the organizers of the International Conference Spatial Cognition for hosting our workshop.

Andreas Nüchter, Oliver Wulf and Kai Lingemann

Fusion of Stereo-Camera and PMD-Camera Data for Real-Time Suited Precise 3D Environment Reconstruction

Klaus-Dieter Kuhnert and Martin Stommel

Institute of Realtime Learning Systems, University Siegen
Hoelderlinstrasse 3, 57068 Siegen, Germany,
{Kuhnert, Stommel}@fb12.uni-siegen.de

Abstract. 3D environment reconstruction is a basic task, delivering the data for mapping, localization and navigation in mobile robotics. We present a new technique that combines a stereo-camera system with a PMD-camera. Both systems generate distance images of the environment but with different characteristics. It is shown that each system compensates effectively for the deficiencies of the other one. The combined system is real-time suited. Experimental data of an indoor scene including the calibration procedure are reported.

1 Introduction

To measure the geometrical structure of the operating environment has been a very fundamental problem in mobile robotics for quite a long time. 2D or 3D laser scanner and stereo-camera arrangements have been investigated thoroughly for indoor and outdoor applications [1][2][3][4][5]. Also techniques with changing active illumination have been employed, e.g. [6][7]. Also combinations of these techniques have been tried [8]. These approaches often lack the necessary speed and robustness for real time navigation and localization. Active techniques suffer from low frame rates. The interpretation of stereo images on the other hand consumes a lot of computing power and does not always deliver stabile results.

Since the advent of PMD-cameras [9] some first tries have been made on person tracking in 3D [10][11] and fusion of the 3D data from the PMD-sensor with the image of a CCD camera [12][13]. We are going to present a newly developed system that combines the distance image stemming from a PMD-camera with one generated by a two camera baseline stereo arrangement.

2 Problem Statement

The reconstruction of the surrounding of a robot from measurements can be characterized with regard to the following objectives:

- A high precision is important for mapping applications and exact movements, e.g. for docking.

- The speed of data acquisition and processing determines the fastest possible motion.
- For complex working tasks of the robot it is necessary to gather the complete 3D information.

Stereo reconstruction typically is computationally intensive and delivers dense depth information only by applying intelligent guessing. It is difficult to obtain robust results on homogenous object surfaces. Periodical structures also cause difficulties. The occlusion and deocclusion especially at object edges often cannot be measured correctly. On the other hand, the precision depends basically on the length of the baseline of the camera setup and can be quite high.

Active techniques need a bright, regularly-reflecting target to work well. Thus, black surfaces with high extinction or specular surfaces cannot be measured correctly. Laser scanners deliver sufficient resolution and due to their widespread application in industry they are mostly quite robust. In contrast, the acquisition rate for depth images is not comparable with standard CCD cameras. Depending on the application, high prices for laser scanners can also be a major drawback.

Therefore a combination of both techniques with each techniques compensating for the measurement flaws of the other one seems favourable. Since for mobile robots a real-time suited combination is necessary, we have chosen the PMD-technology for the active camera.

3 PMD-Camera

The PMD camera directly delivers a depth image in camera centric coordinates by actively illuminating the scene with modulated light.

3.1 General operation

The PMD camera is a solid state lidar-type system. Modulated light is emitted from the active light source and received by a special chip consisting of an array of photo mixer devices (PMD) [14].

The light is generated by two arrays of LEDs modulated with a sinus-wave. The light is reflected by the target and received by a PMD-element. This PMD-element is fed by the original sinus-wave at the electronic side. The received delayed optical signal is mixed with the original electronic signal. This mixing is executed by controlling the charge flow of two coupled photodiodes. As result the phase difference between the original signal and the received signal is measured. From the phase difference the time of flight - and therefore the distance to the target - can be easily computed.

In our experiments we used a 1K PMD-camera with 64*16 Sensor elements. The LEDs emitted at 800 nm in the near infrared. The illuminating light was modulated by 20MHz amplitude modulation, which gives a measurement range of $15\text{m}/2 = 7.5\text{m}$. With a stronger illumination also larger distances could be

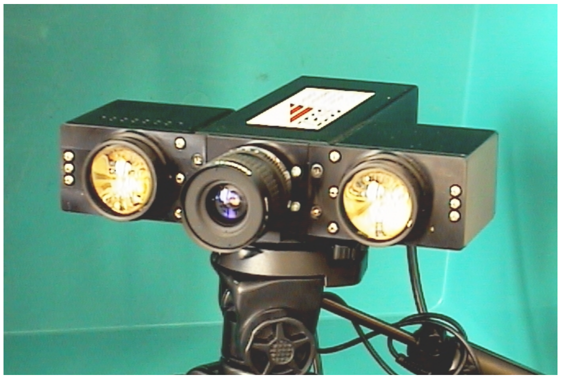


Fig. 1. PMD-Camera

measured but the corresponding phase would not be unique. The construction of the illumination device is mainly limited by the maximal frequency and thermal power dissipation of the LED arrays. On the other hand it has to be eye-save. Practically, with an integration time of 80 ms (12.5 Hz) and a viewing angle of 70.5 degree the maximal measuring distance was about 5m. The integration time can be freely chosen in microsecond steps, but with a smaller integration time the signal/noise ratio decreases and the measurement range will be reduced further.

Our camera has a special circuitry to suppress ambient light by a factor of 100.000. So it can be used indoor and outdoor. This suppression mechanism proved to work robust in our experiments. There is also a model with higher resolution (160*120 PMD elements) but this device can only be utilized indoors because it possesses no ambient light suppression. The camera delivers three images:

- A common intensity image, comparable a CMOS-camera,
- a distance image computed from the phase differences,
- a modulation image showing the modulation-quotient and thus giving a measure for the signal quality of the sensor element.

3.2 Calibration

For each pixel the PMD-camera delivers the distance between the summarized illumination, a target surface element and the sensor element of the array. If the distance is sufficiently large, the illumination can be approximated by a point source near the middle of the camera lens and the camera itself can be approximated by central projection. Furthermore the light source is assumed to be at the centre of projection. These approximations are utilized to transform the measured values into a Cartesian coordinate system.

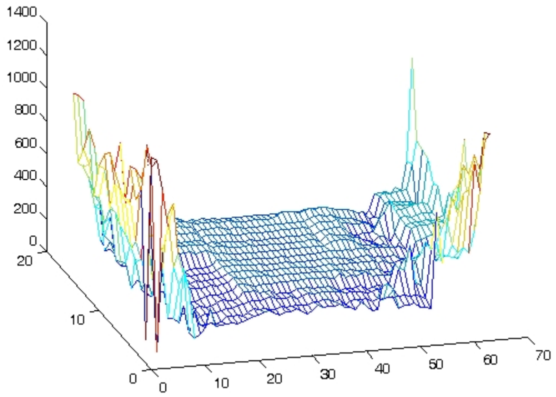


Fig. 2. Distance matrix of a flat surface at a distance of 3m

First the rays from the centre of projection to all pixels are computed. It should be mentioned, that the lateral resolution is not uniform but increases slightly at the borders of the sensor array. The Cartesian representation is then obtained by standard transformation from polar coordinates. Fig. 2 shows the distance matrix in Cartesian coordinates for a flat surface of high reflectance with two nearer marker objects at 3 m. This setup has been used for the calibration process at different distances to the camera.

To measure the phase, the mean value of the middle 5*5 pixels of the phase image was computed for several distances. The relation between phase and distance then was approximated by a linear function (fig. 3). A maximum error of about 3% was measured. Because as an important factor the signal/noise ratio of phase determines the precision, the standard deviation was measured. Illumination decreases quadratically. Therefore, a 2nd order polynomial was fit to the data and it proved to be appropriate (fig. 4). At a first glance the measurement principle seems to allow a precision which would be independent of the distance. But the strong influence of the illumination causes a heavy decay of precision for larger distances.

All these measurements have been executed in the central region because it is the simplest situation. Even here the standard transformation produces asymmetrical results for a plain target surface stemming from a shift of the optical axis by a few pixels. Also lens distortion must be considered at least for angles larger than 20 degree with the optical axis. The first influence was corrected by evaluating the asymmetry and by shifting the coordinate system with $xs = 1.3$ and $ys = 2.5$ pixel. Lens distortion was included in the coordinate transformation.

Even with this enhanced correction some errors remain for measurements in the near region (< 1.5m) and at the border of the array (see fig. 5). In the near region the model of projection is too simple and the source should be

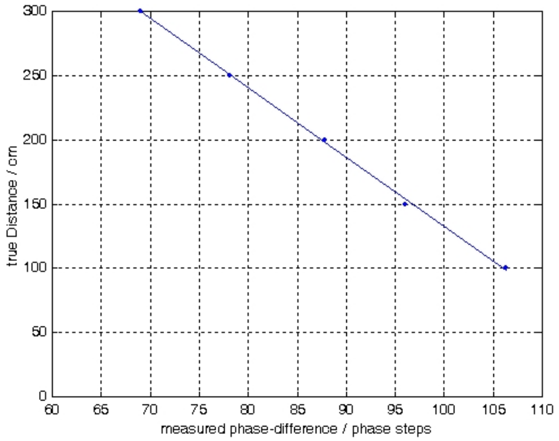


Fig. 3. Distance calibration. The distance function is approximated by the equation $z = -5.4022\varphi + 672.5143$. Dots show the deviation of the samples.

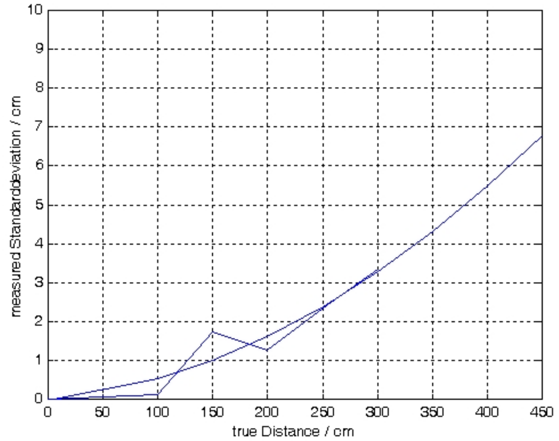


Fig. 4. Measurement precision. The standard deviation is approximated by the function $\sigma = 2.734 \cdot 10^{-5}\varphi^2 + 2.86723 \cdot 10^{-3}\varphi - 4.229692 \cdot 10^{-2}$.

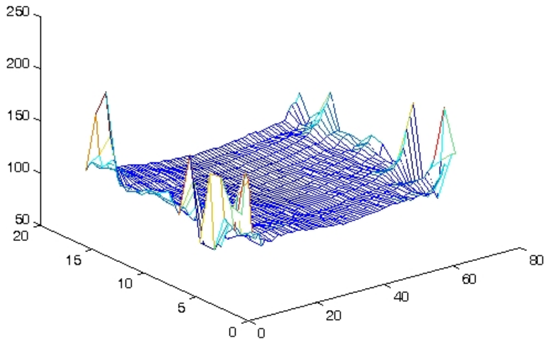


Fig. 5. Distance matrix for a flat surface at a distance of 1m

modelled by a transmitter with finite area to compute the correct near field of the modulation. At the border of the sensor array the illumination is to low to keep the signal/noise ratio sufficiently high. Thus, S/N not only depends on the distance but also on the position in the array. Further the angle between the main direction of the lens and the illumination is about 3 degree, introducing a complicated dependency. Of course the reflectance of the target also influences the precision. The modulation signal observes not exactly the sinus shape causing a non linear phase relation that depends on the signal amplitude. All these influences can be most easily subsumed by observing the modulation image. The modulation ratio is thresholded at 30% and all pixels on these positions are ignored. By this method it can be guaranteed that the precision of the measurement is described in good approximation (20%) by the function of fig. 4.

One very important effect remains to be described. If surfaces at different distances remit light to one PMD-element, a mixed signal will be received. Assuming two surfaces the resulting signal will be a linear combination of the two sinus-waves which are weighted by the reflectance/distance ratio. Especially at occluding edges the phase will be somewhere between the value of the nearer and of the farther surface. Thus, it has to be interpreted with care. The only statement about such a measurement can be: The true value lies between the measurements of its neighbours. This also is an approximation because complex configurations of several surfaces may exist that are projected on one PMD-element. But even for scenes containing several overlapping objects it is a reasonable assumption. Experimentally only situations with very extended objects having high depth differences (e.g. observing a wall nearly parallel to its surface) showed noticeable deviations.

With these considerations the measurement of a scene can be described by two depth maps with a precision better than 5%. One depth map contains the minimal allowable depth, the other one the maximal depth. It would also be possible to describe the measurement by its mean and the estimated standard deviation. But because the distribution of the measurements depends on the distance and because we wanted to build a simple real-time suited algorithm for the fusion of the PMD and the stereo depth map the min/max description was chosen.

The depth maps are created the following way. First the original measurements are transformed and corrected according to the calibration. For each pixel the minimum and the maximum value of the 4 or 8 neighbours are computed and saved in the minimum and the maximum map respectively. Both maps are then corrected by subtracting, respectively adding the appropriate value of the precision by distance function (see fig. 4).

The two maps finally show the resulting range allowed by the constraints of the measurement. Interestingly, the PMD camera delivers high precision at surfaces with homogenous colour and surfaces with homogenous distance. At occluding edges the results are most vague. This is in clear opposition to the behaviour of stereo camera systems. Thus, these two types of systems were combined. They effectively compensate for each other's deficiencies.

4 Stereo Camera System

A camera system with two The Imaging Source DFK 21F04 cameras equipped with Cosmicar/Pentax lenses is used. The cameras provide images with a resolution of 640 by 480 pixels. They are arranged according to the standard stereo geometry with one exception: The yaw angles of the cameras are modified to achieve a wider common field of view. To compensate for different roll and pitch angles of the optical axes the cameras are mounted on separate 3d-adjustable 10mm aluminium plates. Fig. 6 shows the camera system.

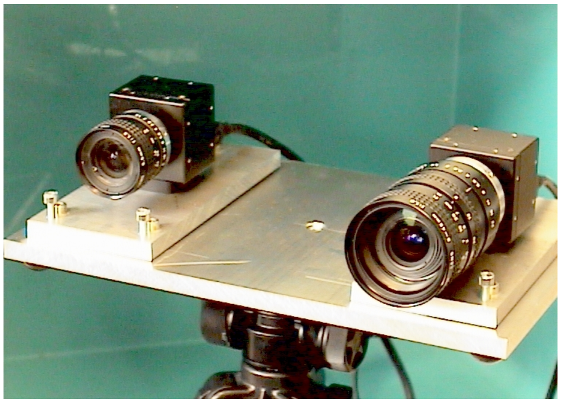


Fig. 6. Stereo camera

4.1 General operation

A point (x, y, z) in 3d-space is projected by the stereo system to the image coordinates (x_l, y_l) in the left image and (x_r, y_r) in the right image. Since in applications for mobile robots often the vertical coordinate y is of minor importance, for a stereo system with parallel optical axes the distance z measured parallel to the optical axis can be computated as

$$z = \frac{b}{x_r/f - x_l/f}. \tag{1}$$

where b denotes the distance between the cameras and f the focal length. Because in our setup the yaw angles - denoted in the equation below as α and β - are modified, the z-coordinate is computed as

$$z = \frac{b}{tan(\alpha + tan^{-1}(-x_r u/f_1)) - tan(\beta + tan^{-1}(-x_l u/f_2))}. \tag{2}$$

The variable u denotes the size of a single pixel. It is given in the data sheet of the Sony ICX098BQ CCD-chip as 5.6 m/pixel. The equation already takes into account that the focal length can be slightly different for both cameras. The corresponding image positions x_r and x_l are found using the "Winner Takes It All" and "Simulated Annealing" stereo matching algorithms based on the implementation of Scharstein and Szeliski [15] and a time optimized version based on the proposals of Sunyoto et al. [16] and Changming Sun [17]. The first one motivated the usage of modern multimedia processor extensions, the second one concerns a recursive subdivision of the stereo images as well as the application of Gaussian smoothing prior to matching.

From the depth map obtained by the application of (2) a minimum and a maximum depth map are computed. These maps give a 95 percent confidence interval for every depth value. They are computed in a straightforward way by subtracting, respectively adding, twice the standard deviation of the measurement error of (2) to the original depth value. Since only structured areas, in particular edges orthogonal to the baseline of the camera setup, result in meaningful depth values, a Sobel operator together with a fixed threshold is used to compute a binary confidence map. For unconfident pixels the minimum depth is set to zero and the maximum depth is set to 10m.

4.2 Calibration

The stereo system is calibrated for a working distance of $1.5-4$m according to the working distance of the PMD-camera. The lens aperture is set to 5.6, the focus to infinity. The lens of the right camera has a fixed focal length of 8mm. The zoom of the left camera lens is adjusted with pixel accuracy to provide images of the same size as the right camera. The aperture angle of the camera was measured as 23.25 degree. The aluminium plates of the stereo setup are adjusted manually to the same roll and pitch angle for both cameras. The remaining error is below 1/100 pixels for the roll angle and 1 pixel for the pitch angle. Since the above mentioned stereo software is based on the assumption that epipolar lines correspond to the rows of the camera images the roll and pitch angle have a direct influence on the quality of the results. The yaw angle was adjusted to centre an object in a distance of 4m in both cameras. The distance between the cameras is 20cm.

After a careful manual adjustment of the stereo system, the parameters b, α, β, f_1 and f_2 were determined in software. To this end a highly textured test object is recorded for distances between 4m and 1.6m. The resulting disparities $x_r - x_l$ are averaged for the object in every image. A genetic algorithm was used to find parameters which minimize the integrated squared error between the distance values predicted by (2) and the measured values. The algorithm was stopped after a sufficient low standard deviation of 0.0077m was achieved. The resulting parameters of our stereo setup are given in tab. 1.

b	0.0024691
α	0.3045662
β	0.3041429
f_1	0.7583599
f_2	0.7580368

Table 1. Camera parameters after calibration

The deviation from the expected values for the camera parameters results from the spread for standard factory models, e.g. CCD chips which are not exactly perpendicular/centred to the optical axes.

5 Fusion Of the Distance Data

The fusion of distance data was performed straightforwardly. Because both 3D sensors are adjusted to a common optical axis, deliver their result in absolute Cartesian coordinates and their relative positions are known in 3D, the coordinate transformation between them is known as well. Thus, the PMD data after correction and calibration have been registered and transformed to the coordinate system of the left stereo camera. Also the lower resolution of the PMD-camera is compensated by replicating the pixel information with appropriate factors in x and y direction. This way matrices of equal size are generated giving the minimal and maximal distances for each sensor. For each pixel the overlapping distance interval is accepted if it exists. If not the measurements are contradictory and will be rejected. The method produces an improved depth map and a map describing the quality i.e. the distance range of each single measurement.

6 Experiments

The experiments were conducted in a corridor of our university under realistic conditions. An example scene is shown in fig. 7. The left image shows an image from the stereo camera. The left part the scene comprises a wall with a door and a poster. A person is standing next to the wall. The right part of the scene consists of a white pillar and a white wall. A low carton is placed in the foreground. Except for the person and the carton the original environment remained unchanged. There are no additional posters or colorful objects which are sometimes found to facilitate the recognition of the environment of a mobile robot. The scene can thus be considered typical for a robot application.

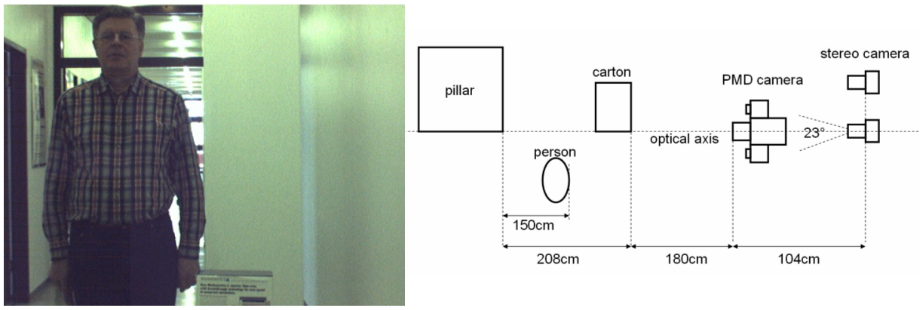
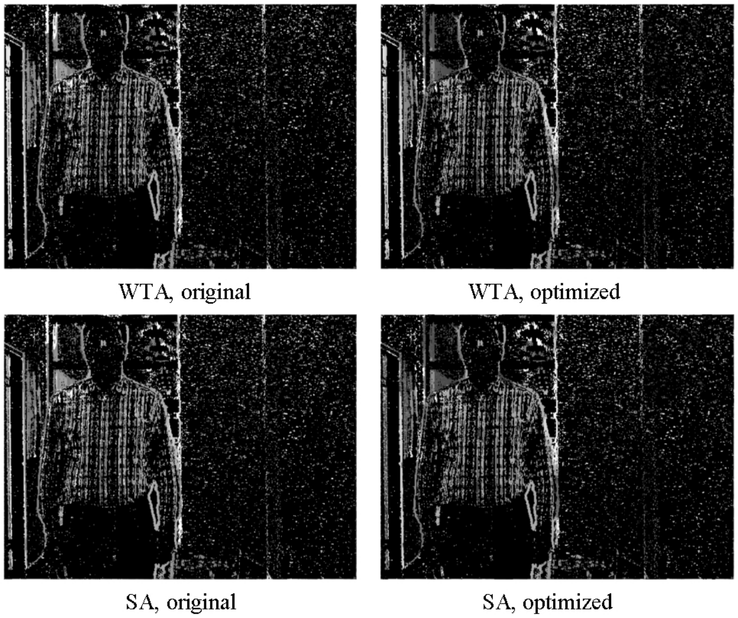


Fig. 7. Experimental setup

The difficulty of the scene for the stereo camera consists mainly in the big poorly structured regions, which make up the biggest part of the scene. A minor problem poses the shirt of the person with its regular pattern. For the PMD camera the object borders pose a difficulty, first because of the abrupt depth discontinuity, and second because of its low spatial resolution.

A schematic of the camera setup is shown on the right side of fig. 7. The stereo camera is placed 104cm behind the PMD camera in a height of 136cm. The PMD-camera stands 30cm lower, so it is not visible in the images of the stereo camera. The distance between the PMD camera and the stereo camera compensates for the different aperture angles. For the detection of stereo correspondences we used an Athlon XP 2800 with 2GB RAM.

Fig. 8 shows the depth maps for different stereo matching algorithms. The stereo system provides good results for the left edge of the pillar, the face and shoulders. The regular pattern of the shirt often causes too high or to low depth values. The right side of the pillar is not recognized because of the low contrast to the background. As expected the results for the walls are sparse and rather random. Concerning the depth values, the simple and fast Winner-Takes-It-All algorithm provides almost the same results as the Simulated Annealing, which is designed for quality instead of speed. In contrast the differences in runtime


WTA, original

WTA, optimized

SA, original

SA, optimized

Fig. 8. Depth maps for different stereo matching algorithms

are quite big as shown in tab. 2. With a computation time in the range of minutes, Simulated Annealing of course is no candidate for the implementation on a mobile robot. But also the comparatively fast Winner-Takes-It-All needs almost 2 seconds per image in the original version, which is still too much for the most applications. By performing the optimizations mentioned in Sec. 4.1, the computation could be accelerated by a factor of four, resulting in a computation time of less than 500 milliseconds for the matching step. This method is thus well suited for real-time applications.

Method	**Processing time [s]**
”Winner takes it all”, original	1.92
”Winner takes it all”, optimized	0.47
”Simulated Annealing”, original	252
”Simulated Annealing”, optimized	67

Table 2. Computation time for stereo matching

Fig. 9 and fig. 10 show the results of the sensor fusion for this example. Fig. 9 gives the deviation between the minimum and the maximum depth of the merged stereo and PMD results. High deviations are indicated by light grey up to

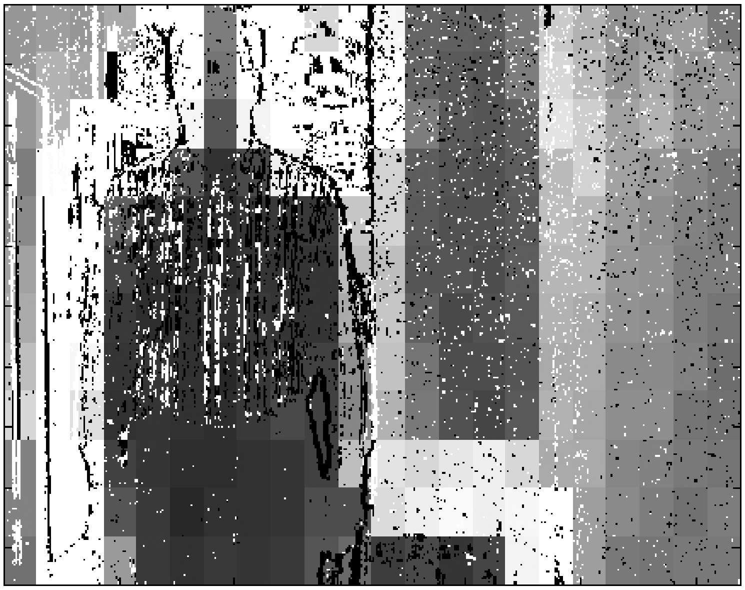


Fig. 9. Maximum deviation of the measurement

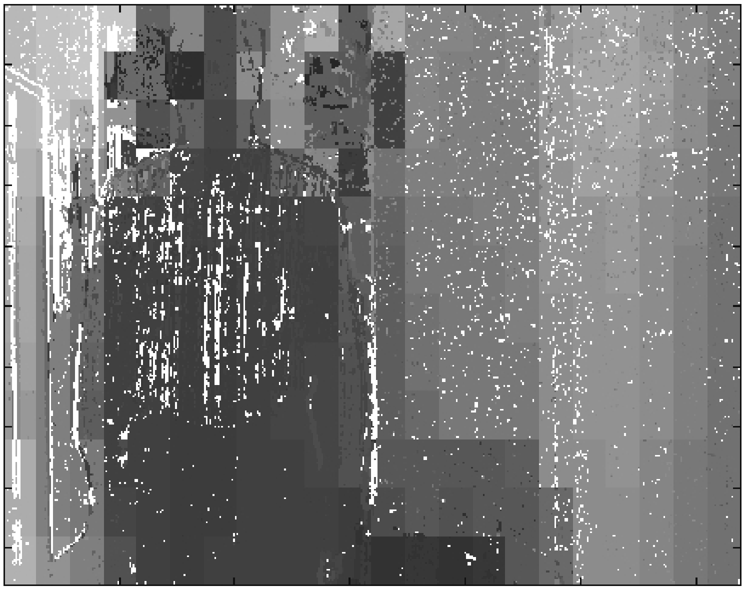


Fig. 10. Absolute Depth

white. White is also used for points without intersection between the confidence intervals of the stereo and PMD measurements, i.e. for points where one or both methods failed to produce reliable results. In comparison to fig. 8 it appears that the PMD-camera provides the most robust depth values for homogenous regions, in particular for the pillar, the nearer part of the wall and the shirt and trousers of the person. The PMD-camera hence provides the values which are missing in the depth map of the stereo system. On the other hand the PMD-camera has a higher inaccuracy in the area of depth discontinuities, e.g. near the head of the person or on the left edge of the pillar. This inaccuracy is caused by the low resolution of the camera. But as can be seen from fig. 9, in these regions stereo matching yields in robust and precise results.

Fig. 10 shows the depth map after sensor fusion. Dark areas mark nearer objects. We obtain a dense array of depth values. In homogenous regions the results mainly stem from the PMD-camera, whereas the results on edges are determined with high precision by the stereo system. There are only a few regions where the two sensors cause differing results. In fig. 10 these pixels are marked in white. They mainly result from ambiguities due to repetitive and other ambiguous structures and occlusion near depth discontinuities.

7 Conclusion

In this paper a new method for the reconstruction of the environment of mobile robots is presented which is based on the fusion of the sensor outputs of a stereo camera and a PMD-camera. It is shown that each technique balances the shortfalls of the other technique while preserving its characteristic benefits. As a result we achieve dense depth maps with both reliable values on homogeneous regions as well as precise and robust values on edges. Because of the optimizations performed on the stereo algorithm and the range measurement in hardware by the PMD-camera, our method is well suited for real-time applications.

8 Acknowledgement

We thank cordially Prof. Dr. Loffeld and Dr. Hartman from the "Zentrum fr Sensorsysteme NRW" for providing the 1K PMD-camera and the support in operating it.

References

1. Schfer, H., Proetzsch, M., Berns, K.: Extension approach for the behaviour-based control system of the outdoor robot RAVON. Autonome Mobile Systeme, 2005.
2. Biber, P.,Andreasson, H., Duckett, T., Schilling, A.: 3D Modeling of indoor environments by a mobile robot with a laser scanner and panoramic camera. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004), 2004.

3. Weiss, C., Zell, A.: Automatic generation of indoor VR-models by a mobile robot with a laser range finder and a color camera. Autonome Mobile Systeme (AMS 2005), Stuttgart, Germany, 2005, Proceedings, pp. 107-113, Springer, 2006.
4. Zhu, Z., Karuppiah, D.R., Riseman, E., Hanson, A.: Adaptive panoramic stereo vision for human tracking with cooperative mobile robots. Robotics and Automation Magazine, Special Issue on Panoramic Robots, 14(10), pp. 69-78, 2004.
5. Porta, J.M., Verbeek, J.J., Krse, B.J.A.: Active appearance-based robot localization using stereo vision. Autonomous Robots 18(1), pp. 59-80, 2005.
6. Kang, S., Webb, J.A., Zitnick, C., Kanade, T.: A multibaseline stereo system with active illumination and real-time image acquisition. Proceedings of the Fifth International Conference on Computer Vision (ICCV '95), June, 1995, pp. 88-93.
7. Viejo, D., Saez, J.M., Cazorla, M.A., Escolano, F.: Active stereo based compact mapping. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Alberta, Canada, August 2005.
8. Sazbon, D., Zalevsky, Z., Rivlin, E.: Qualitative real-time range extraction for preplanned scene partitioning using laser beam coding. Pattern Recognition Letters, 26(11):1772-1781, 2005.
9. http://www.pmdtec.com/
10. Friedland, G., Jantz, K., Knipping, L., Rojas, R.: Experiments on lecturer segmentation using texture classification and a 3D camera. Technical Report B-05-04, Freie Universitt Berlin, Fachbereich fr Mathematik und Informatik, April 2005.
11. Noll, P., Schwab, M., Wiryadi: Sensing people - localization with microphone arrays. Elektronische Sprachsignalverarbeitung (ESSV 2004), Cottbus, September 20-22, 2004.
12. Forkuo, E.K., King, B.: Automatic fusion of photogrammetric imagery and laser scanner point clouds. XXth ISPRS Congress, 12-23 July 2004 Istanbul, Turkey, Proceedings Volume: IAPRS, Vol. XXXV, part B4, pp. 921-926; ISSN 1682-1750.
13. R. Schwarte, personal communication, University Siegen, Germany.
14. Kraft, H., Frey, J., Moeller, T., Albrecht, M., Grothof, M., Schink, B., Hess, H., Buxbaum, B.: 3D-Camera of high 3D-frame rate, depth-resolution and background light elimination based on improved PMD (photonic mixer device)-technologies. OPTO, Nuernberg, May 2004.
15. Scharstein, D., Szeliski, R., Zabih, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In Proceedings of the IEEE Workshop on Stereo and Multi-Baseline Vision, Kauai, HI, Dec. 2001.
16. Sunyoto, H., van der Mark, W., Gavrila, D.M.: A comparative study of fast dense stereo vision algorithms. Proc. of the IEEE Intelligent Vehicles Symposium, Parma, Italy, 2004.
17. Sun, C.: Fast Stereo Matching Using Rectangular Subregioning and 3D Maximum-Surface Techniques. International Journal of Computer Vision. vol.47, no.1/2/3, pp.99-117, May 2002.

3D Environment Cognition in Stereoscopic Robot Teleguide

Salvatore Livatino[1], Filippo Privitera[2]

[1] Medialogy Copenhagen, Aalborg University, Denmark
sal@media.aau.dk

[2] Scuola Superiore di Catania, Italy
fiprivitera@ssc.unict.it

Abstract. The use of 3D stereoscopic visualization in place of 2D viewing may increase telepresence in remote environments, providing a user with higher environment cognition. Works in the literature have demonstrated how stereo vision contributes to improve perception of some depth cues often for abstract tasks, while little can be found about the advantages of stereoscopic visualization in mobile robot tele-guide applications. This work investigates stereoscopic robot tele-guide under different conditions, including typical navigation scenarios and the use of synthetic and real images. This work also investigates how user performance may vary when employing different display technologies. Results from a set of test trials ran on five virtual reality (VR) systems emphasized few aspects which represent a base for further investigation as well as a guide when designing specific systems for telepresence.

1 Introduction

The commonly used 2D display systems suffer of many limitations in robot teleoperation. Among which: misjudgment of self-motion and spatial localization, limited comprehension of remote ambient layout and object size and shape, etc. The above leading to unwanted collisions during navigation, as well as long training periods for an operator. An advantageous alternative to traditional 2D (monoscopic) visualization systems is represented by the use of a stereoscopic viewing. In the literature we can find works demonstrating that stereoscopic visualization may provide a user with a higher sense of presence in remote environments because of higher depth perception, leading to higher comprehension of distance, as well as aspects related to it, e.g. ambient layout, obstacles perception, manoeuvre accuracy, etc. The above conclusions can in principle be extended to tele-guided robot navigation. However, it is hard to find works in the recent literature addressing stereoscopic mobile robot tele-guide, which motivated the authors to focus on this very important application field. In addition, it is not straightforward how stereo viewing would be an advantage for indoor workspaces where the layout, typically man-made, would be simple and emphasizing monocular depth cues such as perspective, texture gradient, etc.

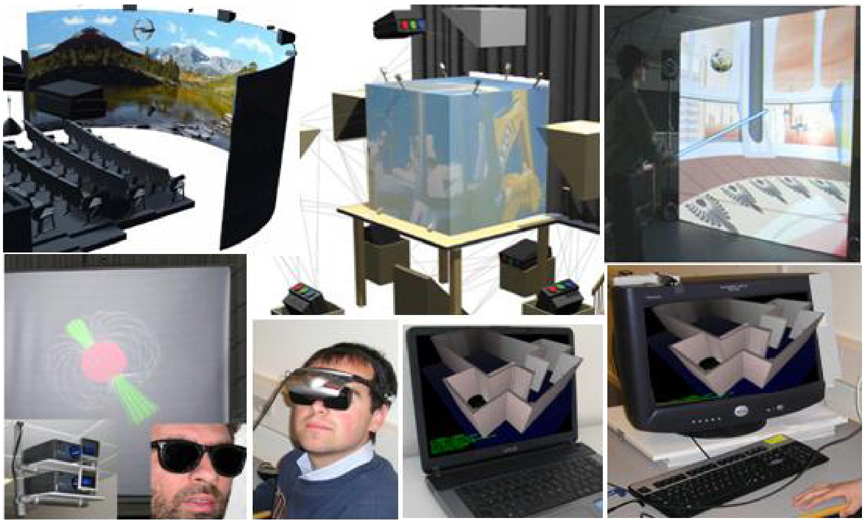


Fig. 1. Virtual Reality facilities at Aalborg University VR Media Lab and Medialogy Copenhagen. Top from left: 160deg Panorama; 6-sided CAVE; 1-sided CAVE. Bottom from left: Small Powerwall with projectors and filters; HMD; 3D Laptop, 3D Desktop.

When analyzing the benefits of stereoscopy, researchers often focus on comparing different depth cues, learning behaviors, etc., but they always run their experimentation trials using one or two specific visualization technologies. A comparison among different VR facilities is uncommon in the literature, despite some works can be found comparing two different systems, e.g. [2], [3]. Nevertheless, depth perception and task performance may greatly vary for different display technologies, providing a user with different sense of presence and interaction capabilities. In addition, display technologies also differ in cost, portability and accessibility. Different display technologies would best fit different application situations. For example, a "light" system, portable and cost-effective, would be required in case of low-range transmission possibility, whereas a larger setup, providing higher immersion, would be more suitable for training purposes.

2 3D Stereo Visualization and Teleoperation

Several systems have been developed for Teleoperation and VR with different display and interaction possibilities, (e.g. [15], [16]). Large displays for immersive presentations, e.g. Powerwalls, Panorama, or systems for individual use but allowing for high interaction, e.g. the CAVE system, [1], or systems with Head Mounted Display (HMD). Figure 1 shows examples. Different technologies have been developed which confirm the fundamental role of stereoscopic visualization for most VR systems. The basic idea supporting stereoscopic visualization is that this is closer to the way we naturally see the world, which tells us about its great potential in teleoperation. Main approaches to 3D stereo visualization may be classified as:

- *Passive Stereo.* Multiplex images in space and they can be sub-divided in: *Anaglyph* (separation based on color filters); *Polarized* (separation based on polarized filters); *Separated Displays* (separation based on different displays very close to user eye as in HMDs).
- *Active Stereo.* Multiplex images in time typically based on *Shutter Glasses*, (LCD shutter panels in synchronization with the visualization display).
- *Autostereoscopic Stereo.* Separates images based on special reflecting sheets laying on the display, or other methods. Do not require goggles.

Different stereoscopic approaches can be used coupled to different display systems. The latter being responsible for the degree of immersion, interactivity, isolation from the surroundings, etc. Among main components:

- *Display Size*, from tiny HMD monitors to large 360deg. panoramic screens.
- *Display Structure*, e.g. flat, curved, table-like, cubic shaped, head mounted.
- *Projection Modality*, LCD/CRT monitors, front/back projected screens.
- *Image Quality*, e.g. resolution, brightness, contrast, color range, refresh rate.

The literature works investigating the benefits of stereoscopy can be classified as either application specific, or abstract tasks with general performance criteria, [2]. In literature test trials often deal with assessing the role of most dominant depth cues, e.g. interposition, binocular disparity, movement parallax, [4], and their consequence to user adaptation to new context (user learning capabilities). The parameters through which assess stereoscopy benefits typically are: item difficulty and user experience, accuracy and performance speed, [4], [5]. Test variables altered during experiments include: changes in monocular cues, texture type, relative distance, etc., other than stereoscopic versus monoscopic visualization. Everybody seems to agree that stereoscopic visualization presents the necessary information in a more natural way, which facilitates all human-machine interaction [5]. In particular, stereoscopy improves: comprehension and appreciation of presented visual input, perception of structure in visually complex scenes, spatial localization, motion judgement, concentration on different depth planes, perception of surface materials. The main drawback, which have yet prevented large application, is that users are called to make some sacrifices, [7]. A stereo view may be hard to "get right" at first attempt, hardware may cause crosstalk, misalignment, image distortion, and all this may cause eye strain, double images perception, depth distortion.

Most of the benefits of stereoscopy may affect robot tele-guide. Among the conclusions gathered from the literature: "most tele-manipulation tasks require operators to have a good sense of the relative locations of objects in remote world", [5]; "stereopsis gives better impression of tele-presence and of 3D layout", [8], [9]; "binocular disparity and movement parallax are important contributors to depth perception", [4]; "a robot in a dangerous environment can be controlled more carefully and quickly when the controller has a stereoscopic view", [14].

3 Robot Tele-Guide and 3D Visualization Technologies

It is proposed to investigate benefits of stereoscopic viewing based on the analysis of few factors typically described as predominant, (e.g. [10], [3]), i.e. depth relationships and motion perception. Two categories of user studies are proposed:

- *Aptitude Tests.* To assess user's ability in estimating egocentric distance and self-motion when using stereoscopic visualization under static conditions or passive (computer controlled) motion. Proposed trials concern "Egocentric Distance", (the user stands in front of a corridor while he/she is asked to estimate the egocentric distance to the far-end wall-plane); and "Self-Motion", (the user is driven along a corridor while is asked to estimate robot speed).
- *Interactive Tests.* To assess user's ability in estimating relative and egocentric distance when using stereoscopic visualization under dynamic conditions or user controlled motion. Proposed trials concern "Collision Avoidance", (the user drives along a narrow corridor avoiding making collisions against the walls); and "Access Width". (the user is asked to estimate access width of visible doorways).

The outcome of the above experimentation is in terms of: *measurement accuracy*, (directly provided by the user as answers to questionnaires, or inferred based on number of collisions registered); and *task-completion time*, (recorded in some trials).

At the Aalborg University we have a large variety of state-of-the-art VR facilities, which represents a formidable testing ground for the proposed investigation (Figure 1 shows the VR facilities). In particular, for our investigation we have considered:

- *3D Laptop.* Medium Notebook, 15in high-res display. Passive anaglyph.
- *3D Desktop.* 21in CRT high-res monitor. Passive anaglyph / active shutters.
- *Small Powerwall.* Front projected 1.5m x 1.5m silver screen, 2 high-res projectors. Passive stereo with polarized filters.
- *1-sided CAVE.* Cost-effective rear-projected 2.5m x 2.5m screen, 1 low-cost low-res high-freq projector. Passive anaglyph / active shutters, [11].
- *Head Mounted Display.* 2x0.59in OLED LCDs 800x600. Separated displays.

User tested the systems observing both simulated and real stereoscopic images. The simulated images were rendered from a graphical model which was constructed as it would be extracted from an elevation map from an onboard laser rangefinder. The real images were recorded with a stereo camera setup for mobile robots expected onboard. We assess systems capabilities for different display technologies asking test users to report about their experience through questionnaires. In particular, questions are grouped into five judgement categories: *adequacy to application*, *realism*, *immersion*, *3D impression*, *viewing comfort.*

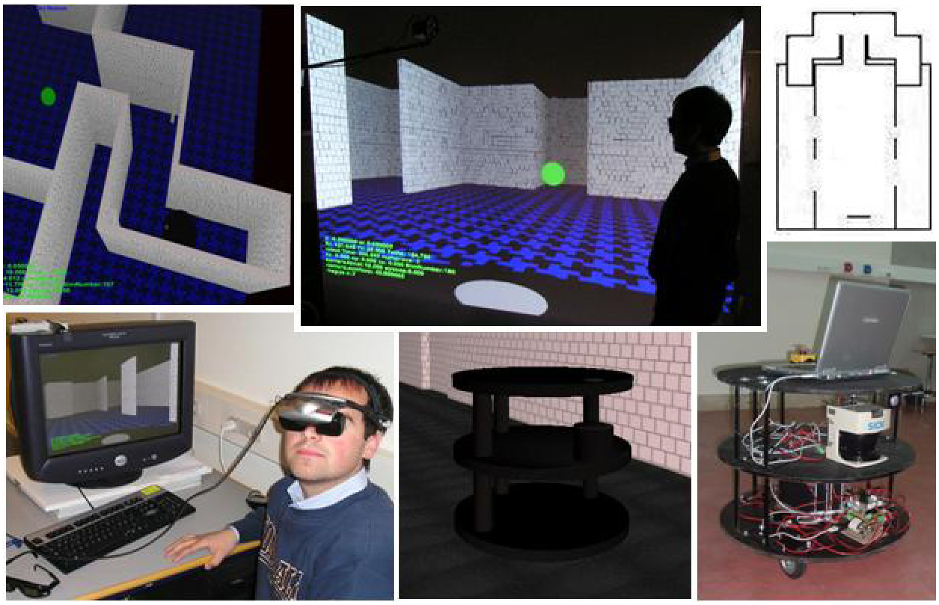


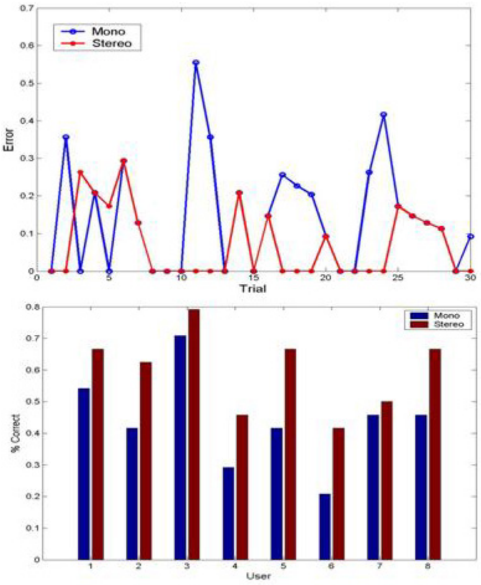
Fig. 2. The simulated workspace. Top: 2 images on the CAVE and workspace map. Bottom: a moment during comparative tests of 3D Desktop and HMD; simulated and real robot (the Morduc system at DIEES, University of Catania, Italy).

4 Testing

Figure 2 shows the simulated workspace. During the experimentation we altered: *Illumination*, *Texture*, *Planes depth*, *Access width*, *Robot speed*. Concerning testing with real stereoscopic images, the users were asked to observe 4 different stereo videos on the VR facilities. The cameras provided low resolution color images suitable for low-bandwidth transmission. Figure 4 shows a stereo snapshot from a recorded video.

After some pilot studies, 15 users were asked to run the Aptitude and the Interactive trials. An error-rate index was calculated based on user answers and ground truth [6]. The Aptitude trials consisted of 2 test trials (distance and motion). In the Interactive trials users were asked to drive through a path which combined the ”Collision Avoidance” and ”Access Width” tests. Figure 2 shows a general workspace map for the Interactive trials.

Under stereoscopic visualization users perform better the Aptitude ”Egocentric Distance” tests and the Interactive ”Access Width” tests. Variance analysis with repeated measures showed a main effect of stereo viewing on percentage of correct answers: F=5.38 and p=0.0388 (egocentric distance); F=5.33 and p=0.0368 (access width). Figure 3 shows accuracy of a typical run, and the percentage of correct answers for a representative pool of users. The highest improvement was obtained on the 3D Desktop. We believe this may due to the lower 3D impression sometime provided in the CAVE for positive parallax,



Synthetic Images		Adequacy	Realism	Immersion	3D Impression	Comfort
3D Laptop Console	*Anaglyph*	4.2	2.5	2.5	5.8	2.5
3D Desktop Console	*Anaglyph*	4.2	2.5	3.3	6.7	3.3
	Shutters	5	3.3	4.2	7.5	5
1-sided CAVE	*Anaglyph*	5.8	5	3.3	6.7 / 5.8	1.6
	Shutters	6.7	5.8	7.5	7.5 / 5.8	6.7
HMD	*Separated*	5	4.2	5	6.7 / 5.8	5.8
Small Powerwall	*Polarized*	6.2	5.8	6.7	7.5	7.5

Video Images		Adequacy	Realism	Immersion	3D Impression	Comfort
3D Laptop Console	*Anaglyph*	1.7	3.3	2.5	5.8	2.5
3D Desktop Console	*Anaglyph*	2.5	2.5	4.2	3.3	2.5
	Shutters	5	4.2	4.2	6.7 / 7.5	5
1-sided CAVE	*Anaglyph*	5.8	5.8	4.2	5.8 / 5	8.3
	Shutters	6.7	6.7	6.7	7.5 / 5.8	6.7
HMD	*Separated*	5	5	5	6.7	6.7

Low (1-4) Medium (5-7) High (8-10)

Fig. 3. Left: inaccuracy of a typical run from the Aptitude Egocentric Distance tests (top) and percentage of correct answers for a representative pool of test users for the Interactive Access Width (bottom). Right: results for the comparative tests with synthetic images (top) and stereoscopic videos (right). Cells with 2 values represent judgment for negative (left) and positive (right) parallax.

(far-end wall-planes appeared "compressed"). The benefits of stereoscopy when estimating robot self-motion were instead not significant. This seemed to agree with the theory of Hubona et al., [12], with motion saturating the visual system so that stereopsis would not be relevant. The Interactive "Collision Avoidance" trials did not provide significant results.

The results of the comparative tests are shown in figure 3. We can observe that users believe that larger visualization screens provide higher Realism and Adequacy of depth cues in robot teleguide than other VR systems. This goes along with Demiralp et al. considerations , [2], telling that "looking-out" tasks (i.e. where the user views the world from inside-out as in our case), require users to use more their peripheral vision. Larger screens provide higher Immersion, (as expected). Interestingly, the sense of immersion drops in case of passive anaglyph, mostly justified by eye-strain arising from rear-projection (screen alters colors causing high crosstalk). It may surprise the reader that most users claim a higher 3D Impression with 3D Desktop rather than with the CAVE. In particular, this concerned the behind display impression (positive parallax). Confirmation that 3D Desktop perceived 3D Impression can be large, can be found in the work of Jones et al., [13]. In our case the range of perceived depth represents a large workspace portion for small screens than for larger. The highest Comfort

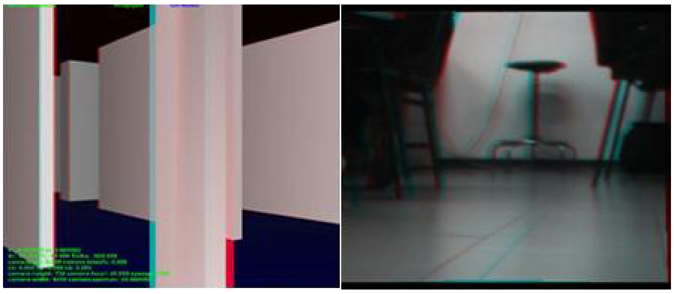


Fig. 4. Passive anaglyph stereo images: graphical simulation (left), real image (right).

judgement is assigned to the Small Powerwall, as confirmation of the benefits of front-projection and polarized filters.

The real videos were in general best appreciated in the CAVE, then on the HMD, and then on the 3D Desktop with active stereo. A higher Realism was felt by the users compared with synthetic images (except in case of Anaglyph stereo), while Adequacy was generally worse. The Immersion category gets the same scores obtained for synthetic images but scores for passive anaglyph are higher (probably a consequence of higher realism). 3D Impression is generally worse. The viewing comfort is typically worse too. It is best for HMD and CAVE with shutters and almost unacceptable for CAVE with anaglyph. Figure 3 shows average scores for the different judgment categories.

5 Conclusion

The proposed work investigated the role of 3D stereoscopic visualization in applications related to mobile robot teleguide. Results from a set of test trials ran on five different VR systems emphasized few differences which represented a base for further investigation. The stereoscopic viewing improves performance in case of estimation of egocentric and related distance, while it does not show significant improvements in case of self-motion perception compared to other depth cues. A main purpose of the proposed investigation was also the comparison of different systems, which characteristics were described in terms of adequacy to application, realism, immersion, 3D impression, and viewing comfort. We hope that our comments on those aspects can support system design for specific applications in Telerobotics (to be considered in relation to cost and portability). Future investigation will concern with a deeper analysis of the Interactive tests and the extension of our tests to large VR displays such as Panoramas, Powerwalls, and the 6-sided CAVE.

References

1. Cruz-Neira, C., Sandin D.J., DeFanti T.A.: Surround-screen projection-based virtual reality: the design and implementation of the CAVE SIGGRAPH93

2. Demiralp, C., Jackson, C.D., Karelitz, C., Zhang S., Laidlaw D.: CAVE and Fishtank Virtual-Reality Displays: A Qualitative and Quantitative Comparison IEEE Transactions on Visualization and Computer Graphics. **12(3)** (2006)
3. Kasik, D.J., Troy, J., Amorosi S.R., Murray, M.O., Swamy, S.W.: Evaluating Graphics Displays for Complex 3D Models IEEE Computer Graphics and Applications **22**, (2002)
4. Naeplin, U., Menozzi, M.: Can Movement Parallax Compensate Lacking Stereopsis in Spatial Explorative Tasks? Elsevier DISPLAYS **22** (2006)
5. Drascic, D.: Skill Acquisition and Task Performance in Teleoperation using Monoscopic and Stereoscopic Video Remote Viewing Human Factors Society (1991)
6. Gaggioli, A., Breining, R.: Perception and Cognition in Immersive Virtual Reality Communications Through Virtual Technology: Identity Community and Technology in the Internet Age (2001)
7. Sexton, I., Surman, P.: Stereoscopic and Autostereoscopic Display Systems IEEE Signal Processing Magazine (1999)
8. Bocker, M., Runde, D., Muhlback, L.: On the Reproduction of Motion Parallax in Videocommunications 39th Human Factors Society (1995)
9. Geiser, A.: Ergonomische Grundlagen fur das Raumsehen mit 3D Anzeigen Dissertation ETH Zurich (1994) Nr.10656
10. Matlin, M.W., Foley, H. J.: Sensation and Perception Allyn and Bacon, 1983)
11. Livatino, S.: Designing a Virtual Reality Game for the CAVE Eurographics IC '06, Catania, Italy (2006)
12. Hubona, G.S., Shirah, G.W., Fout, D.G.,: The Effects of Motion and Stereopsis on Three-Dimensional Visualization Int. J. Human-Computer Studies **47** (1997)
13. Jones G. Lee, D., Holliman, N., Ezra, D.: Perceived Depth in Stereoscopic Images Proc. 44th Human Factors Society San Diego, USA, (2000)
14. Bimber, O.: What Will It Be? Computer Graphics Siggraph'05, USA (2005)
15. T. Boult. DOVE: Dolphin Omni-directional Video Equipment. IC on Robotics and Automation, 2000.
16. R. Ott, M. Gutierrez, D. Thalmann, F. Vexo. Advanced VR Technologies for Surveillance and Security. IC on VR Continuum and Its Applications (VRCIA), 2006.

Navigating Mobile Robots with 3D Laser Data in Real Time

Oliver Wulf and Bernardo Wagner

{wulf,wagner}@rts.uni-hannover.de

Two main problems in mobile robotics are localization and navigation. It is common practice to solve these problems based on 2D range sensor data. But there are situations in real world indoor and outdoor environments where 2D data is not sufficient. Examples are overlapping and negative obstacles as well as well as cluttered environments and uneven terrain. To overcome these problems we are using 3D laser range scanner for environment perception. Our focus lies on the extraction of obstacle and landmark information from 3D sensor data. As our 3D navigation system is used to control mobile robots all algorithms need to be computable in real time on a moving platform. This presentation gives an overview of our 3D laser scanner and the concept of Virtual 2D Scans. But the focus of our presentation lies on the demonstration of real-world robotic applications using 3D environment perception.

3D Laser Scanner

The 3D scanner consists of a standard 2D laser range finder (Sick LMS 291-S14) and an additional servo drive (RTS/ScanDrive). This approach is common praxis [1][2][3] as there is no commercial 3D laser scanner available that meets the requirements of mobile robots. The specialties of our ScanDrive are a number of optimizations that are made to allow fast scanning. One mechanical optimization is the split-ring connection for power and data. This connection allows continuous 360° scanning without the accelerations and high power consumption that are typical for panning systems. Even more important than the mechanical and electrical improvements is the precise synchronization between the 2D laser data, servo drive data and the wheel odometry. Having this good synchronization, it is possible to compensate systematic measurement errors and to measure accurate 3D point-clouds even with a moving robot. The data output of the laser scanner is an set of 3D points given in a robot centered Cartesian coordinate system.

3D point

$$p = (x, y, z) \in R^3$$

3D point-cloud

$$L = \left\{p^{(i)}\right\}_{i=1\ldots n} = \left\{(x, y, z)^{(i)}\right\}_{i=1\ldots n}$$

The data rates that are used in our experiments are between 240x181 points (n=43440) scanned in 3.2s and 90x181 points (n=16290) in 1.2s (angle resolution 4° horizontal x 0.5° vertical).

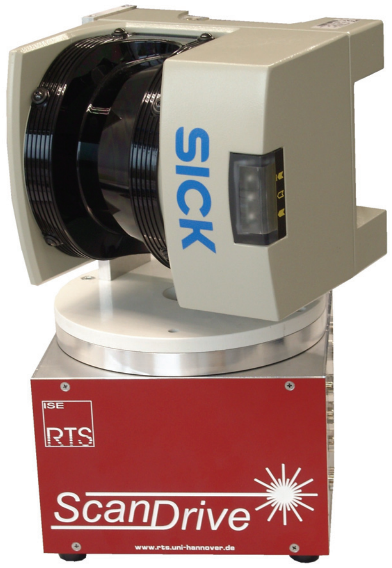


Fig. 1. 3D Laser Scanner RTS/ScanDrive

Virtual 2D Scans

The 3D point-clouds that are acquired by the 3D scanner contain detailed information about the surrounding environment. Because 3D point-clouds are raw data representations, they include redundant information and many measurement points which are not needed for localization and mapping. Approaches which use this raw data for scan matching and full 3D modeling are computational expensive. If the goal is to localize or navigate a mobile robot, these full 3D algorithms are not efficient. The use of Virtual 2D Scans is more efficient as it aims to reduce the amount of data without loosing information that is essential for mobile robot localization. The reduced data sets can afterwards processed with computationally less expensive 2D matching and SLAM algorithms. The data representation that is chosen for Virtual 2D Scans is similar to the data that can be measured directly with a 2D laser range sensor.

2D point

$$\lambda = (x, y) \in R^2$$

Virtual 2D Scan

$$\Omega = \left\{\lambda^{(i)}\right\}_{i=1...m} = \left\{(x, y)^{(i)}\right\}_{i=1...m}$$

For this reason existing 2D scanners can be replaced with a more capable 3D perception systems using existing 2D localization, SLAM and path planning algorithms.

The Virtual 2D Scan is processed in two steps. The reduction of the 3D point-cloud:

$$V \subset L \text{ with } |V| = m \text{ and } m << n .$$

And the projection onto the horizontal plane:

$$f_v : V \to \Omega_v \text{ with } f_v : (x, y, z) \to (x, y)$$

The process if reducing the 3D point-cloud is depending on the application and the environments. Different algorithms and heuristics can be found in [4], [5] and [6].

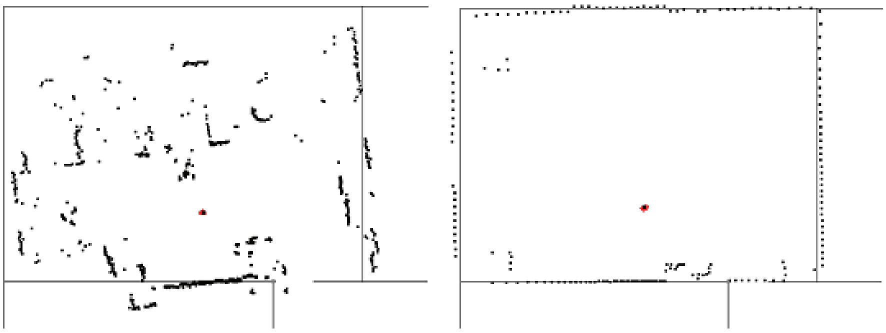


Fig. 2. Comparison of a 2D Scan (left) and a Virtual 2D Scan (right). The Virtual 2D Scan is optimized to find walls in a cluttered room

Application Overview

The presentation gives an overview over a number of demo applications using the 3D perception system. The demo applications feature the full mobile robot navigation circle including localization, mapping and path planning. Robot operation is demonstrated in cluttered indoor environments [4], urban outdoor environments [5][7] and industrial halls [6][8].

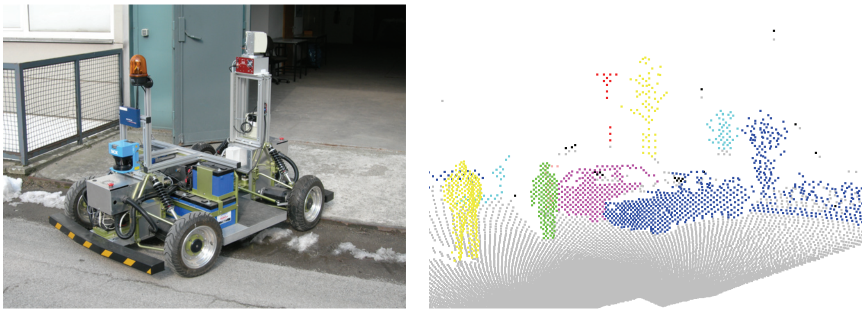


Fig. 3. Mobile Robot RTS/Dora (left), 3D point-cloud (right)

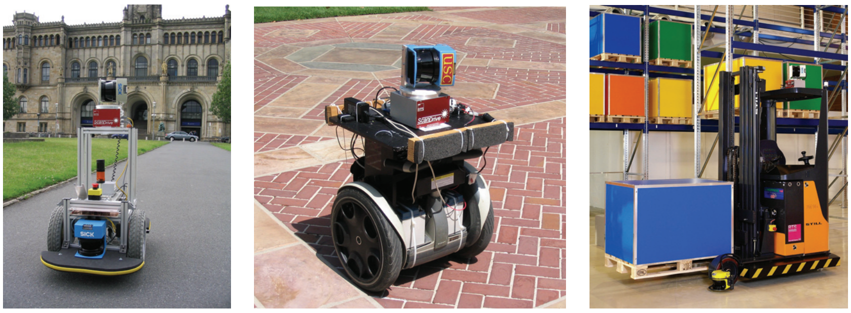


Fig. 4. RTS/MoRob-Kit (left), Segway (center), RTS/STILL Robotic Fork-Lift (right)

References

[1] Hähnel, D. and W. Burgard (2002). “Map building with mobile robots in populated environments”, *International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland

[2] Nüchter A., Surmann H., Lingemann K. and J. Hertzberg (2004). “6D SLAM with application in autonomous mine mapping”, *International Conference on Robotics and Automation*, New Orleans, USA

[3] Wulf O. and B. Wagner (2003) “Fast 3D scanning methods for laser measurement systems”, *International Conference on Control Systems and Computer Science*, Bucharest, Romania

[4] Wulf O., Arras K. O., Christensen H. I. and B. Wagner (2004) “2D mapping of cluttered indoor environments by means of 3D perception”, *International Conference on Robotics and Automation*, New Orleans, USA

[5] Wulf O., Brenneke C. and B. Wagner (2004) “Colored 2D maps for robot navigation with 3D sensor data”, *International Conference on Intelligent Robots and Systems*, Sendai

[6] Wulf O., Lecking D.. and B. Wagner (2006) “Robust self-localization in industrial environments based on 3D ceiling structures”, *International Conference on Intelligent Robots and Systems*, Beijing, China

[7] B. Wagner (2006) “Technical Paper – Team RTS University of Hannover”, *1st European Land Robot Trial (ELROB)*, Hammelburg, Germany

[8] Lecking D., Wulf O., Viereck V., Tödter J. and B. Wagner (2006) “The RTS-STILL Robotic Fork-Lift”, *EURON Technology Transfer Award*, Palermo, Italy

Detecting Useful Landmarks for Visual SLAM

Simone Frintrop, Patric Jensfelt, and Henrik Christensen

Computational Vision and Active Perception Laboratory (CVAP),
CSC, Kungliga Tekniska Högskolan (KTH), Stockholm, Sweden
Email: {frintrop/patric/hic}@csc.kth.se

Abstract. In this paper, we introduce a new method to automatically detect useful landmarks for visual SLAM. Landmarks are detected by a biologically motivated attention system. Various experimental results on real-world data show that the landmarks are useful with respect to be tracked in consecutive frames and to enable closing loops.

1 Introduction

An active area of research in mobile robotics is *simultaneous localization and mapping (SLAM)*, where a map is autonomously constructed of the environment. SLAM for indoor settings based on laser scanners is today considered a mature technology. However, the laser scanner is much too expensive for many applications. Therefore and because of the high amount of information offered by camera images, the focus within the community has shifted towards using visual information instead [1, 4, 6, 5]. One of the key problems in SLAM is *loop closing*, i.e. the ability to detect when the robot is revisiting some area that has been mapped earlier.

To perform SLAM, landmarks have to be detected in the environment. A key characteristic for a good landmark is that it can be reliably detected, this means that the robot is able to detect it over several frames. Additionally, it should be redetectable when the same area is visited again, this means the detection has to be stable under viewpoint changes. Often, the landmarks are selected by a human expert or the kind of landmark is determined in advance. Examples include localization based on ceiling lights [10]. As pointed out by [9], there is a need for methods which enable a robot to choose landmarks autonomously. A good method should pick the landmarks which are best suitable for the current situation.

In this paper, we suggest to use a computational visual attention system [2] to choose landmarks for visual SLAM. The advantage of this method is that it determines globally which regions in the image discriminate instead of locally detecting predefined properties like corners. We evaluate the usefulness of the attentional regions of interest (ROIs) and show that ROIs are easily tracked over many frames even without using position information and are well suited to be redetected in loop closing situations.

The application of attention systems to landmark selection has rarely been studied. Two existing approaches are [7], in which landmarks are detected in

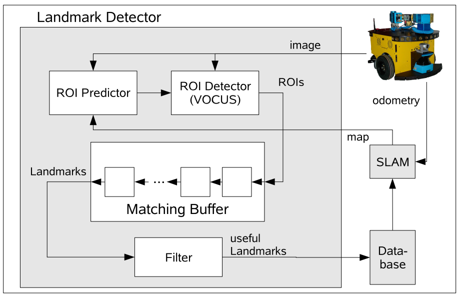


Fig. 1. Attentional landmark selection for visual SLAM

hand-coded maps, and [8], in which a topological map is build. The only approach we are aware of which uses an approach similar to a visual attention system for SLAM, is presented in [6]. They use a saliency measure based on entropy to define important regions in the environment primarily for the loop closing detection in SLAM. The map itself is built using a laser scanner though.

2 System Overview

In this section, we give an overview over the architecture in which the landmark detection is embedded (see Fig. 1). More details on the architecture can be found in [3]. The main components are the *robot* which provides camera images and odometry information, a *landmark detector* to create landmarks, a *database* to store the landmarks, and a *SLAM module* to build a map of the environment.

When a new frame from the camera is available, it is provided to the landmark detector. The landmark detector consists of a *ROI predictor* which predicts the position and appearance of landmarks depending on previous frames and on position estimations from the SLAM module, a *ROI detector* which redetects predicted ROIs and finds new ROIs based on the visual attention system VOCUS, a *matching buffer* which stores the last n frames, performs matching of the ROIs in these frames and creates landmarks from matched ROIs, and a *filter module* which filters out low quality landmarks.

The landmarks which pass the filter are stored in the database and provided to the SLAM module which performs the estimate of the position of landmarks and integrates the position into the environmental map. To detect old landmarks, the landmark position estimates from the SLAM module are used to narrow down the search space in the database.

The ROI detector consists mainly of the biologically motivated attention system VOCUS which detects regions of interest similar to the human visual system. The computations are based on detecting strong contrasts and uniqueness of features for the features intensity, orientation, and color (details in [2, 3]). The matching of ROIs between frames is based on the similarity of the ROIs (depending on the size of the ROI and a feature vector which describes the appearance of the ROIs) and on the prediction, i.e. the expected position of the

ROI considering the movement of the robot (details about the matching buffer in [3]). Details about the robot and the SLAM architecture can be found in [5].

3 Experiments and Results

The experiments were performed on a sequence of 658 images, obtained in a hallway by a robot driving 2.5 times in a circle. This setting is especially well suited to investigate loop closing situations, since most of the visited regions appear again after a while. The experiments consist of two parts: first, we show how the matching of ROIs can be used for tracking over consecutive frames and investigate the matching quality with and without position prediction. Second, we show how the matching of ROIs can be used in loop closing situations.

Tracking First, we investigate the matching of ROIs between consecutive frames. To investigate the matching by similarity independently from proximity, we first only consider matching of the ROIs based on the size of the ROIs and the similarity of the feature vector and second combine it with the position prediction.

We investigated how the number of landmarks, the length of the landmarks (i.e. the number of associated ROIs), and the quality of the matching (nb. false matches) depends on the choise of the matching threshold δ (Tab. 1, top). It shows that the number of landmarks increases when δ increases as well as the length of the landmarks, but on the other hand there are also significantly more false matches.

Finally, we performed the same experiments with additional position prediction (Tab. 1, bottom). It shows that the false matches are avoided and the matching is robust even for higher values of δ. This means that for a high threshold combined with position prediction, a higher amount of landmarks is achieved while preserving the tracking quality.

Loop closing In the next experiment, we investigate the matching quality in loop closing situations. We proceed as follows: for each ROI that is detected in a new frame, we match to all ROIs from all landmarks that occurred so far. We used a tight threshold for the matching ($\delta = 1.7$) and assumed that no position information is available for the ROIs. This case corresponds to the “kidnapped robot problem”, the most difficult version of loop closing. If additional position information from odometry is added, the matching gets more precise, false matches are avoided and the threshold might be chosen higher.

In these experiments, the system attempted to perform 974 256 matches between ROIs (all current ROIs were matched to all ROIs of all landmarks that were detected so far). Out of these possible matches, 646 were matched, 575 (89%) of these matches were correct. Fig. 2 shows two examples of correct matches and one of a false match.

The experiments show that the attentional ROIs are useful landmarks which are both successfully tracked over consecutive frames and suitable to be redetected after visiting an area again from a different viewpoint.

matching without prediction:

Threshold	# landm.	# ROIs	# ROIs/Landmark (av)	# false m.
1.7	62	571	9.2	1
2.0	73	730	10.0	7
2.5	88	957	10.9	15
3.0	102	1109	10.9	42
5.0	130	1568	12.0	147

matching with prediction:

Threshold	# landm.	# ROIs	# ROIs/Landmark (av)	# false m.
1.7	61	566	9.3	0
2.0	73	724	9.9	0
2.5	88	955	10.9	0
3.0	98	1090	11.1	0
5.0	117	1415	12.1	0

Table 1. Matching of ROIs for different thresholds δ

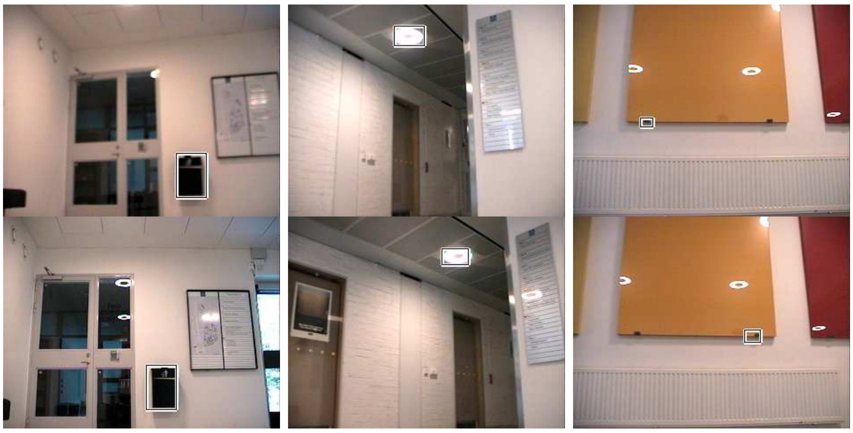


Fig. 2. Matching results in loop closing situations: the first row shows a ROI in a current frame, the second row shows a previously seen ROI from the database. This situations usually occurred after the robot drove a circle and revisited the same area again. Two correct matches (left, middle) and one false match (right) are shown.

4 Conclusion

In this paper, we have presented a method for landmark selection based on a biologically motivated attention system which detects salient regions of interest. The matching of regions shows to be successful not only in short-term tracking but also in loop closing. The detection and matching method has been tested in a SLAM scenario to demonstrate the basic performance for in-door navigation.

Acknowledgment

The present research has been sponsored by the European Commission through the project NEUROBOTICS (EC Contract FP6-IST-001917) and the Swedish Foundation for Strategic Research through its Center for Autonomous Systems. The support is gratefully acknowledged.

References

1. Davison, A. J. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In: Proc. of the ICCV (2003).
2. Frintrop, S. VOCUS: A Visual Attention System for Object Detection and Goal-directed Search. PhD thesis Bonn Germany (2005). Published 2006 in Lecture Notes in Artificial Intelligence (LNAI), Vol. 3899, Springer Verlag.
3. Frintrop, S., Jensfelt, P. and Christensen, H. Attentional Landmark Selection for Visual SLAM. In: Proc. of the International Conference on Intelligent Robots and Systems (IROS '06)(accepted) (2006).
4. Goncavles, L., di Bernardo, E., Benson, D., Svedman, M., Ostrovski, J., Karlsson, N. and Pirjanian, P. A visual front-end for simultaneous localization and mapping. In: Proc. of ICRA (2005) 44–49.
5. Jensfelt, P., Kragic, D., Folkesson, J. and Björkman, M. A Framework for Vision Based Bearing Only 3D SLAM. In: Proc. of ICRA'06 (2006).
6. Newman, P. and Ho, K. SLAM-Loop Closing with Visually Salient Features. In: Proc. of ICRA'05 (2005) 644–651.
7. Nickerson, S. B., Jasiobedzki, P., Wilkes, D., Jenkin, M., Milios, E., Tsotsos, J. K., Jepson, A. and Bains, O. N. The ARK project: Autonomous mobile robots for known industrial environments. *Robotics and Autonomous Systems* **25** (1-2, 1998) 83–104.
8. Ouerhani, N., Bur, A. and Hügli, H. Visual Attention-based Robot Self-Localization. In: Proc. of ECMR (2005) 8–13.
9. Thrun, S. Finding Landmarks for Mobile Robot Navigation. In: Proc. of the 1998 IEEE International Conf. on Robotics and Automation (ICRA '98) (1998).
10. Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Cremers, A., Dellaert, F., Fox, D., Hähnel, D., Rosenberg, C., Roy, N., Schulte, J. and Schulz, D. Probabilistic algorithms and the interactive museum tour-guide robot Minerva. *Int'l J. of Robotics Research* **19** (11, 2000).

Monocular-vision based SLAM using line segments

Thomas Lemaire and Simon Lacroix

LAAS-CNRS
Toulouse, France
{thomas.lemaire, simon.lacroix}@laas.fr

Abstract. This paper presents a new approach for using line segments in vision-based SLAM. This work is based on a previous algorithm which tackles the initialisation problem in EKF-SLAM using an initial sum of Gaussians for point features. Here, the supporting line of a segment is represented with its Plücker coordinates, a representation well adapted for the initialisation and estimation processes. Results on real data are presented.

1 Problem statement

Vision is becoming more and more popular in the SLAM community: cameras can easily be embedded on a robot, they gather a lot of useful data, and allow the development of 3D SLAM approaches. But a single camera does not provide the *depth* information of the perceived features (vision based SLAM is often referred to as *bearing-only SLAM*): this raises the need to develop a specific feature initialisation algorithm for the commonly used extended Kalman filter framework.

Several initialisation methods for point visual features have recently been proposed. The *delayed* methods accumulate data about the landmarks until a Gaussian estimate is computed [1,2], whereas the *un-delayed* methods add an estimate of the landmark since the first observation [3,4]. Recently, a special *inverse depth* representation of a 3D point has been used [5,6]. The key properties is that with this representation the observation function is nearly linear in the neighbourhood of the camera pose: hence the initialisation procedure does not suffer the tedious linearization step of the EKF, as an acceptable Gaussian estimate of the inverse-depth representation of the point can directly be computed.

As compared to point features, line segment features have interesting properties to build a representation of the environment: they provide more information on the structure of the environment, and are more invariant with respect to viewpoint changes. Moreover, numerous line segments can be detected in structured or semi-structured environments. The two algorithms presented in [5,6] have been adapted to introduce segments in the map [7,8]. But using a camera moving forward at low frame rates leads to dramatic divergence because the conditions for the observation function to be linear are not satisfied.

In this work, an extension to the multi-hypotheses monocular SLAM algorithm [2] for 3D line-segments is presented. It is based on a well-adapted representation for the 3D line segments and a careful initial probability density function (PDF) that approximates the state.

2 Segments in monocular SLAM

2.1 Segment representation.

3D line. The choice of a good representation for 3D lines is essential for the overall estimation algorithm. Among the various minimal or non-minimal possibilities, we selected one that is suited with the projection model of a 3D line in a camera: this projection defines the 3D plane Π going through the focal point and the segment. The Plücker coordinates of a 3D line are very well adapted, since the plane Π is directly represented by its normal $\underline{n}$ (a more detailed description of the Plücker coordinates can be found in [9]):

$$L_{(6\times1)} = \begin{pmatrix} n = h.\underline{n} \\ \underline{u} \end{pmatrix} \tag{1}$$

h is the distance of the line to the origin, and $\underline{u}$ is the direction of the line. This 6-dimension non-minimal representation is constrained by the two relations:

$$\begin{cases} ||u|| = 1 & \text{(normalisation)} \\ n \cdot \underline{u} = 0 & \text{(Plücker constraint)} \end{cases} \tag{2}$$

These constraints are considered in the estimation process (section 2.2).

Segment extremities. Line segments extraction algorithms do not produce segments with stable extremities. This is the reason why the observations and the stochastic map only contain information related to the *supporting line*. However, extremities are very important to give a higher level of significance to the map. In our framework the extremities of a segment are represented by deterministic coordinates in a frame attached to the line.

2.2 Stochastic estimation

Initialisation. As noted before, a single observation gives a measure of the plane Π: with observation noise modelled as a centred Gaussian distribution, this first observation gives a Gaussian estimate of the vector $\underline{n}$. Two other magnitudes do not have a Gaussian estimate: the depth and the direction of the 3D line. As in [2,4] in which the points depth is initialised as a PDF defined by a sum of Gaussians, the depth of the 3D line is also approximated by a geometric sum of Gaussians, defined according to a geometric progression (the depth hypotheses lay in the direction defined by the vector $\underline{g}$). The direction $\underline{u}$ of the line is in the plane Π, and is approximated considering its angle θ in Π. This orientation θ is approximated by a uniform sum of Gaussians. Figure 1 shows the geometric construction of these hypotheses. Also, typical Gaussian sum for depth and orientation and a 3D view of the resultant hypotheses are shown Figure 2.

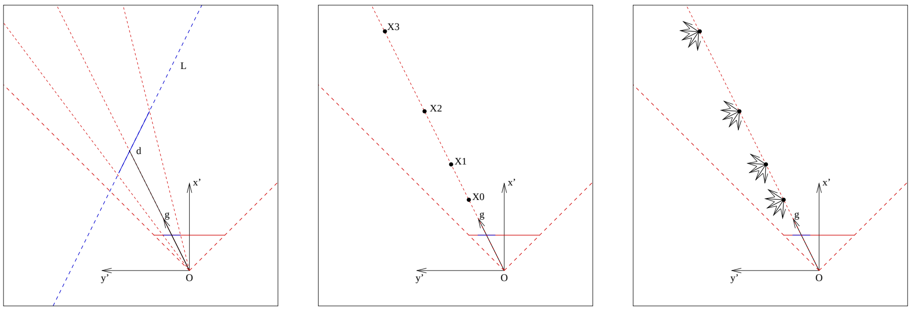


Fig. 1. (Ox'y') corresponds to the plane Π, in red the image plane projection in Π and the camera aperture. From left to right: *(a)* projection of line L and definition of the vector $\underline{g}$, *(b)* the set of points X_i that correspond to the mean of the Gaussians whose sum represents the PDF on the line depth, and *(c)* the set of lines considered to initialise the estimation process.

As in [2], as the robot moves each new observation of the line is used to update the likelihood of all the initial hypotheses, less likely hypotheses are pruned, and the initialisation process is iterated until a single hypotheses remains: it is then added in the stochastic map, and its parameters are further estimated using the SLAM extended Kalman filter.

Kalman update and constraints. Once an hypothesis has been selected and added to the stochastic map, there is no guarantee that the Kalman updates will not break the two constraints of equation 2. To ensure the correctness of the Plücker representation, the technique of *smooth constraints* is used (see [10])

3 Experimental results

Parameters definition in simulation. Our algorithm is driven by several parameters. Table 1 gives the list of the parameters and the values used in these experiments. The values are set according to their physical meaning and according to extensive simulation tests that have been conducted.

Image segments matching. To ensure robust and reliable segment matches, we rely on the Harris interest points matching algorithm presented in [11]: to each segment are associated the closest matched interest points, and segment matches are established according to a hypothesis generation and confirmation paradigm. This simple process has proved to yield outlier-free matches (figure 3), even for large viewpoint changes, which is very helpful to associate landmarks when closing loops.

Results. Results on an image sequence acquired with an iRobot ATRV are presented Figure 3. Here the robot moved along a 5 meters diameter circular trajectory, and the robot odometry is used to feed the prediction step of the SLAM

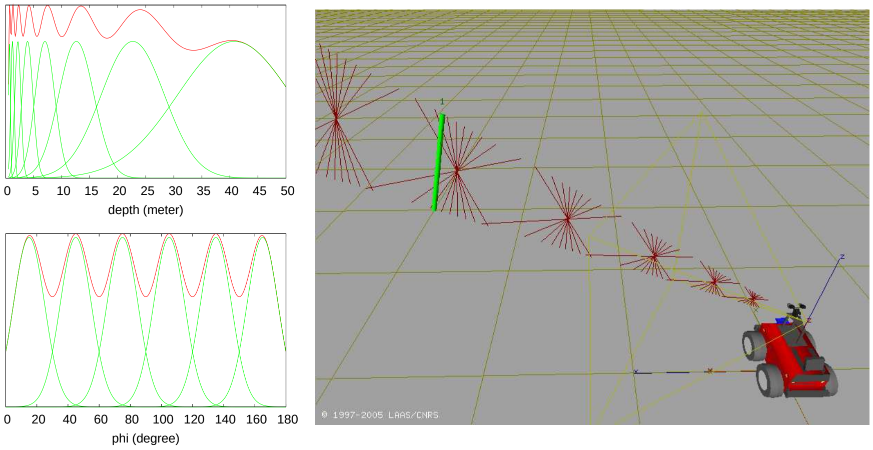


Fig. 2. Left: geometric sum of Gaussian in the range $[0.5, 50]$ with $\alpha = 0.25$ and $\beta = 1.8$, uniform sum of Gaussians in the range $[0, 180]$ with $\sigma = 10$ and $k_\sigma = 1.5$. Right: 3D view: in green the real segment, in red the set of hypothesis.

process. The camera is looking sidewards to the centre of the circle where two boxes have been put. At the end of the process, horizontal and vertical edges estimates have consistent Plücker coordinates (within the 3σ bounds).

Note that depending on the trajectory of the robot, some landmarks are never initialised because their coordinates are not well observed. With point features, these are the points that lie in the direction of the motion of the camera. Similarly, with segment features, the lines which are contained in the plane on which the camera is moving cannot be initialised.

4 Discussion

The presented approach is well suited to build an environment representation based on 3D line segments from a single camera within a SLAM framework.

parameter	description	value
β_d	rate of the geometric series	1.3
α_d	ratio between mean and standard-deviation	0.2
σ_ϕ	standard-deviation of each hypothesis	4°
k_{σ_ϕ}	where 2 consecutive Gaussians meet in a fraction of σ_ϕ	1.3
τ	threshold to prune bad hypothesis	10^{-2}
α_c	initial constraint noise factor	0.1
th_c	threshold on relative strength to trigger constraint application	100

Table 1. Parameters of the algorithm.

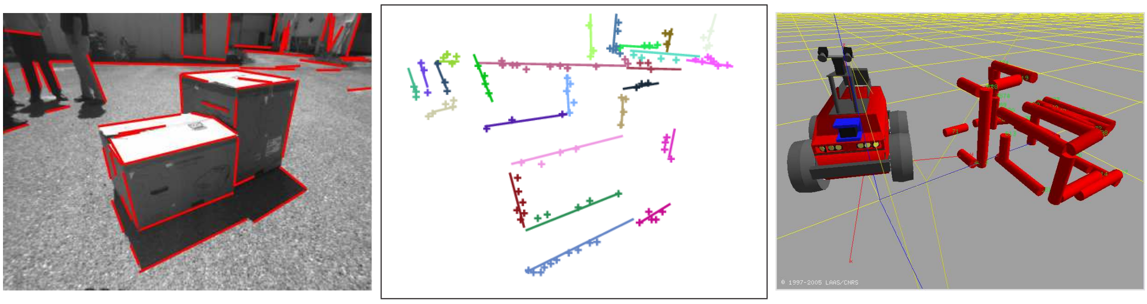


Fig. 3. Left: segments extracted in the image. Centre: segments matched on points matches basis. Right: a 3D model obtained during a trajectory of about 16 meters around the two boxes.

In the light of simulations, it appears that the application of the Plücker constraint is not absolutely necessary. This is due to the correlations which are computed in the initialisation procedure. Nevertheless, this has to be verified in long terms experiments.

Future efforts certainly include more work on the perception side, especially the stability of the segment detection and extraction, and the matching algorithm. Segment features are more invariant than points when the viewpoint changes. This property can be very useful for loop closing and also for multi robots cooperative localisation and mapping.

References

1. Davison, A.: Real-time simultaneous localisation and mapping with a single camera. In: Proc. International Conference on Computer Vision, Nice. (2003)
2. Lemaire, T., Lacroix, S., Solà, J.: A practical 3d bearing only slam algorithm. In: IEEE International Conference on Intelligent Robots and Systems. (2005)
3. Kwok, N.M., Dissanayake, G.: An efficient multiple hypothesis filter for bearing-only slam. In: IROS. (2004)
4. Solà, J., Devy, M., Monin, A., Lemaire, T.: Undelayed initialization in bearing only slam. In: IEEE International Conference on Intelligent Robots and Systems. (2005)
5. Eade, E., Drummond, T.: Scalable monocular slam. In: CVPR 2006. (2006)
6. Montiel, J.M.M., Civera, J., Davison, A.J.: Unified inverse depth parametrization for monocular slam. In: RSS 2006. (2006)
7. Eade, E., Drummond, T.: Edge landmarks in monocular slam. In: BMVC. (2006)
8. Smith, P., Reid, I., Davison, A.: Real-time monocular slam with straight lines. In: BMVC. (2006)
9. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press, ISBN: 0521540518 (2004)
10. Geeter, J.D., Brussel, H.V., Schutter, J.D., Decreton, M.: A smoothly constrained kalman filter. IEEE Transactions on Pattern Recognition and Machine Intelligence **19** (1997) 1171–1177
11. Jung, I.K., Lacroix, S.: A robust interest point matching algorithm. In: International Conference on Computer Vision, Vancouver (Canada) (2001)

Using Treemap as a Generic Least Square Backend for 6-DOF SLAM

Udo Frese

FB 3 Mathematik und Informatik, SFB/TR 8 Spatial Cognition,
Universität Bremen,

Abstract. Treemap is a generic SLAM algorithm that has been successfully used to estimate extremely large 2D maps closing a loop over a million landmarks in 442ms. We are currently working on an open-source implementation that can handle most variants of SLAM. In this paper we show initial results demonstrating 6-DOF feature based SLAM and closing a simulated loop over 106657 3D features in 209ms.

1 Introduction

Simultaneous Localization and Mapping (SLAM) has been a central topic in mobile robotics research for almost two decades by now [1]. Most of the literature is concerned with generating a 2D map with a sensor moving in the plane (3-DOF). Only in the last few years the problem of generating a 3-D map with a sensor moving in 3D space (6-DOF) has received considerable attention [2–5]. Such a system has important applications, for instance rescuing victims from the remains of a collapsed building. So we expect that 6-DOF SLAM will be a growing research area, in particular with the recently emerging 3D cameras.

Many 2D SLAM articles have been concerned with efficiency in estimating large maps ([6] for an overview). In 6-DOF SLAM the efficiency discussion has mainly focused on the first stages of processing in particular on 3D scan matching [7]. 3D maps always contain a lot of data but up to now little attention has been paid to 3D maps, that are by magnitudes larger than the sensor range.

We contributed the treemap algorithm [8] to this discussion in 2D SLAM. It is designed for computing least square estimates for very large maps efficiently. Using treemap we were able to demonstrate closing a simulated loop with one million landmarks in 442ms [9]. On the one hand, the treemap algorithm is sophisticated but also complicated. On the other hand, it is fairly general mainly estimating random varianbles of arbitrary meaning. Hence our current project is to develop an open source implementation that – as an *implementation* – can be used to perform most variants of SLAM including 2D, 3D, features and/or poses, and visual SLAM. This workshop paper reports intermediate results showing a simulated 6-DOF SLAM experiment (3D features, no odometry) that uses the same implementation as our previous million-landmarks (2D features, odometry, marginalized poses) experiment. By building on the efficiency of treemap as a backend, we where able to close a loop over $n = 106657$ 3D features in 209ms.

2 Local and Global Challenges for 6-DOF SLAM

Many challenges currently addressed in 6-DOF SLAM concern the first stages of sensor processing: matching 3D scans, finding reliable features, matching them, rejecting outliers, filtering range images or handling bearing-only initialization. These problems are local in the sense that they affect only that part of the map that is currently observed by the sensor. In contrast there is also the question how this local information and its uncertainty affects the global map. The most prominent situation is certainly closing a loop when the local information that closes the loop leads to back-propagation of the error along the loop. The key point, as we have argued in [6, §12], is that the local uncertainty is small but complex and depends on the actual sensor and the detailed circumstances of observation, whereas the global uncertainty is mostly the composition of local uncertainties, i.e. it is large, rather simple and dominated by the map's geometry.

This insight motivates our treemap approach. In the past it has motivated the design of the treemap algorithm itself that exploits this locality. And now it motivates our idea that many different SLAM variants (2D / 3D, features and/or poses, with/without odometry) can be solved by a specific local preprocessing plus treemap as a global least-square backend. From this perspective a large map is mainly a matter of computation time and hence our goal is:

> Whenever you can formulate your SLAM approach in a least-square framework such that it works for small maps, you can use treemap as a backend to make it work for large maps.

The following section gives the overall idea of the treemap algorithm from a general probabilistic perspective. A more extensive presentation as well as the concrete Gaussian implementation can be found in [9, 8].

3 The Treemap Algorithm

Imagine the robot is in a building that is virtually divided into two parts A and B. Now consider: *If the robot is in part A, what is the information needed about B?* Only few of B's features are involved in observations with the robot in A. All other features are not needed to integrate these observations. So probabilistically speaking, the information needed about B is only the marginal distribution of features of B also observed from A conditioned on observations in B.

The idea can be applied recursively by dividing the building into a binary tree of regions and passing probability distributions along the tree (Fig. 1). The input to treemap are observations assigned to leaves of the tree. They are modeled as distributions $p(X|z_i)$ of the state vector X, i.e. of feature positions and robot poses, given some measurement z_i. With respect to the motivating idea, nodes define local regions and super-regions. Formally, however, a node $\mathbf{n}$ just represents the set of observations assigned to leaves below $\mathbf{n}$ without any explicit geometric definition. During the computation, intermediate distributions $p^M_{\mathbf{n}}$ and $p^C_{\mathbf{n}}$ are passed through the tree and stored at the nodes, respectively.

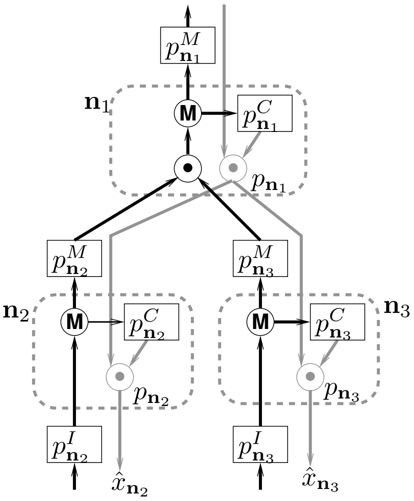


Fig. 1. Data flow view of the probabilistic computations performed by treemap. The leaves store the input $p^I_{\mathbf{n}}$. During updates (black arrows), a node $\mathbf{n}$ integrates ($\odot$) the distributions $p^M_{\mathbf{n}_\swarrow}$ and $p^M_{\mathbf{n}_\searrow}$ passed by its children. The result is factorized (Ⓜ) into a marginal $p^M_{\mathbf{n}}$ passed up and a conditional $p^C_{\mathbf{n}}$ stored at $\mathbf{n}$. To compute an estimate (gray arrows), each node $\mathbf{n}$ receives a distribution $p_{\mathbf{n}_\uparrow}$ from its parent, integrates ($\odot$) it with the conditional $p^C_{\mathbf{n}}$, and passes the result $p_{\mathbf{n}}$ down. In the end, estimates $\hat{x}_{\mathbf{n}}$ are available at the leaves.

A feature is passed in distributions $p^M_{\mathbf{n}}$ from the leaves where it is involved, up to the least common ancestor of all these leaves. There it is marginalized out and finally stored in $p^C_{\mathbf{n}}$. So the distribution $p^M_{\mathbf{n}}$ passed by a node contains those features involved in leaves below $\mathbf{n}$ but also involved above $\mathbf{n}$. An estimate is computed recursively down the tree. A node receives an estimate for the features in $p^M_{\mathbf{n}}$ and combines it with the conditional $P^C_{\mathbf{n}}$ stored at $\mathbf{n}$ to compute an estimate for the features marginalized out there which is in turn passed down.

Figure 2 shows the Bayesian justification for this approach: For each node $\mathbf{n}$ the features $X[\mathbf{n}\colon \not\downarrow\uparrow]$ and measurements $z[\mathbf{n}\colon \not\downarrow\uparrow]$ only above $\mathbf{n}$ are conditionally independent from the features $X[\mathbf{n}\colon \downarrow\not\uparrow]$ and measurements $z[\mathbf{n}\colon \downarrow\not\uparrow]$ only below $\mathbf{n}$ given the features $X[\mathbf{n}\colon \downarrow\uparrow]$ involved at the same time above and below $\mathbf{n}$.

It should be noted, that the process of passing distributions along the tree is exact. The only approximations are linearization when computing the input $p^I_{\mathbf{n}}$ and sparsification needed when old robot poses are marginalized out.

4 Different Variants that could be Supported

In 2D SLAM there are mostly two variants used. The first is *consistent pose estimation* where 3-DOF poses are estimated from 3-DOF links derived from odometry and scan matching. The second is the classical variant with 2D point features (sometimes also 2-DOF lines) and 3-DOF poses where old poses are marginalized out. This requires *sparsification*, an additional approximation to preserve locality during marginalization. We used this variant in our million-landmarks experiment.

In 6-DOF SLAM more variants are possible. Using 3D scan matching [2] one can perform 6-DOF consistent pose estimation. In contrast to 2D SLAM usually no odometry is available. This gives rise to a very simple variant, where poses are marginalized out immediately. Since there is not odometry, sparsity is

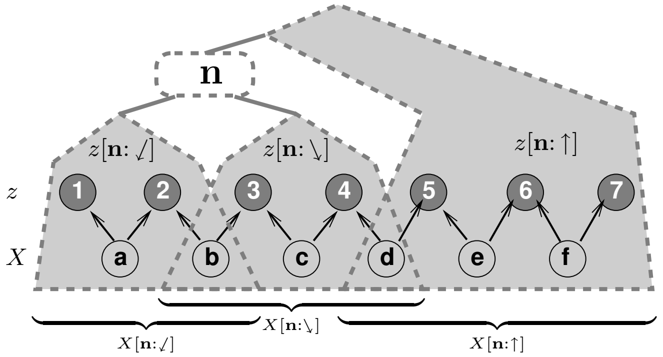

n
$z[\mathbf{n}\colon\swarrow]$
$z[\mathbf{n}\colon\searrow]$
$z[\mathbf{n}\colon\uparrow]$
z
X
1
2
3
4
5
6
7
a
b
c
d
e
f
$X[\mathbf{n}\colon\searrow]$
$X[\mathbf{n}\colon\swarrow]$
$X[\mathbf{n}\colon\uparrow]$

Fig. 2. Bayesian View. In this example, observations $z_{1\ldots7}$ provide information about the features $X_{\mathsf{a}\ldots\mathsf{f}}$. The arrows and circles show this probabilistic input as a Bayes net with observed nodes in gray. The dashed outlines illustrate the view of a single node $\mathbf{n}$. It divides the tree into three parts, *left-below* $\swarrow$, *right-below* $\searrow$ and *above* $\uparrow$. Hence the observations z are disjointly divided into $z[\mathbf{n}\colon\swarrow] = z_{1\ldots2}$, $z[\mathbf{n}\colon\searrow] = z_{3\ldots4}$ and $z[\mathbf{n}\colon\uparrow] = z_{5\ldots7}$. The corresponding features $x[\mathbf{n}\colon\swarrow] = X_{\mathsf{a}\ldots\mathsf{b}}$, $x[\mathbf{n}\colon\searrow] = X_{\mathsf{b}\ldots\mathsf{d}}$ and $x[\mathbf{n}\colon\uparrow] = X_{\mathsf{d}\ldots\mathsf{f}}$ however overlap ($X[\mathbf{n}\colon\swarrow\searrow] = X_{\mathsf{b}}$, $X[\mathbf{n}\colon\uparrow\searrow] = X_{\mathsf{d}}$). The key insight is that $X[\mathbf{n}\colon\downarrow\uparrow] = X[\mathbf{n}\colon\swarrow\uparrow \vee \searrow\uparrow] = X_{\mathsf{d}}$ separates the observations $z[\mathbf{n}\colon\downarrow]$ and features $X[\mathbf{n}\colon\downarrow\not\uparrow]$ below $\mathbf{n}$ from the observations $z[\mathbf{n}\colon\uparrow]$ and features $X[\mathbf{n}\colon\not\downarrow\uparrow]$ above $\mathbf{n}$, so both are conditionally independent given $X[\mathbf{n}\colon\downarrow\uparrow]$. The notation follows [8].

maintained and no sparsification is needed. Essentially, this means, that each set of observations is converted into relative information on the involved 3D point features. This variant is presented in the experiments here.

It has a major limiation. Without odometry a small sensor blackout or too little overlap between observations will disintegrate the map because no information links the involved two poses anymore. Inertial sensors can help by providing relative orientation (gyros) and absolute inclination (accelerometers). This is the pendant of classic SLAM with 3D point features and 6-DOF poses marginalized out later. Still, with orientation-odometry only, consecutive observations must share one feature. Yet another variant uses the accelerometers as translation-odometry. But when acceleration is integrated the result is relative velocity not relative position, so the poses must be augmented by 3D velocity (9-DOF total).

With a monocular camera [4], no distance can be measured. So, while consistent pose estimation can use the 5-DOF links arising from matching two images [10], additional information is needed for the overall scale. In a feature based approach this leads to the corresponding problem of bearing-only initialization.

5 Towards an Open Source Implementation

We believe that these different variants can all be based on treemap as the same least square backend. The main difference lies in the size of the vectors representing features and poses, in different Jacobians for linearization, and in the way an initial estimate can be computed for linearization. Another difference

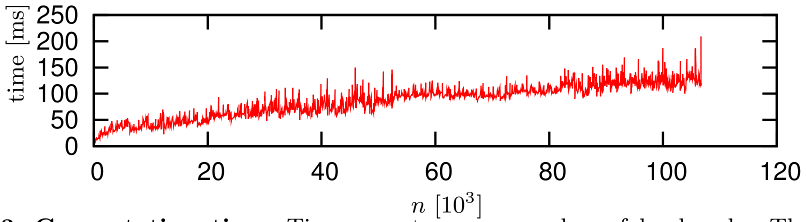


Fig. 3. Computation time. Time per step over number of landmarks. The final increase in computation time happens after closing the loop.

is the control policy: Which observations are combined in one leaf? Are old poses marginalized out? When is sparsification used for the sake of efficiency? When are Jacobians recomputed from the original nonlinear observations?

All 6-DOF SLAM variants share the problem of parameterizing 3D orientation. We extend a technique by Castellanos [11] to 3D and use the product

$$Q = Q_0 \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \approx Q_0 \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix}$$

of a fixed orientation Q_0 and three Euler rotations the angles of which are the random variables estimated. Q_0 is initialized with the current estimate so the Euler angles only parameterize the small *perturbation* of the orientation and are far from singularity. Hence they are always linearized at $\alpha = \beta = \gamma = 0$ and the linearization has the simple form shown above.

This technique also allows to reduce the linearization error caused by error in the robot orientation. The distributions passed from a nodes children are rotated according to the current estimate before multiplying them (Fig. 1, $\odot$) [8].

The goal of our current project is to implement all the different SLAM variants. The treemap backend is already finished with a sufficiently generic interface so we were able to implement a driver for feature based 2D SLAM [9] in 2100 lines of C++ code and a driver for feature based 6-DOF SLAM without odometry in 1200 lines of code. The following section shows results for the latter.

6 Experimental Results

In our simulated experiment the robot moves through a 20 story building with features on the rooms walls (Fig. 4). Then it crosses a bridge on the 19^{th} floor into another 20 story building and maps that building too. Finally it returns to the starting position and closes a loop over all feature. The overall map has $n = 106657$ features and $m = 5319956$ observations from $p = 488289$ poses. Poses are not represented in the map. Computation time was at most 209ms (Fig. 3).

7 Conclusion and Outlook

We have demonstrated that that the treemap algorithm in the same generic implementation can be used to solve both 2D and 3D feature based SLAM

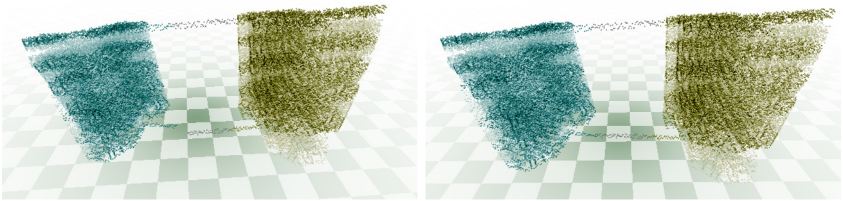


Fig. 4. 6-DOF SLAM map. before and after closing the large loop (between the two building on the ground level) over all $n = 106657$ features. A 3D animations of the growing 3D map and the previous 2D million-landmarks simulation can be downloaded from our web site `www.informatik.uni-bremen.de/~ufrese/`.

(without odometry) with high efficiency. Future work includes implementing the remaining SLAM variants, integrating a solution to the bearing-only initialization problem and implementing a 3D variant of the rotation technique used to reduce linearization error.

We then plan to publish the implementation as an open source library.

References

1. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (2005)
2. Surmann, H., Nüchter, A., Lingemann, K., Hertzberg, J.: 6D SLAM - preliminary report on closing the loop in six dimensions. In: Proceedings of the 5th Symposium on Intelligent Autonomous Vehicles, Lissabon. (2004)
3. Miro, J.V., Dissanayake, G., Zhou, W.: Vision-based SLAM using natural features in indoor environments. In: Proceedings of the 2005 IEEE International Conference on Intelligent Networks, Sensor Networks and Information Processing. (2005)
4. Davison, A., Cid, Y., Kita, N.: Real time SLAM with wide angle. In: Proc. IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon. (2004)
5. Ohno, K., Tadokoro, S.: Dense 3D map building based on LRF data and color image fusion. In: Proceedings of the International Conference on Intelligent Robots and Systems. (2005) 1774–1779
6. Frese, U.: A discussion of simultaneous localization and mapping. Autonomous Robots **20**(1) (2006) 25–42
7. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, Quebec City. (2001) 145 – 152
8. Frese, U.: Treemap: An $O(\log n)$ algorithm for indoor simultaneous localization and mapping. Autonomous Robots (2006) to appear.
9. Frese, U., Schröder, L.: Closing a million-landmarks loop. In: Proceedings of the IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems, Beijing. (2006)
10. Eustice, R., Singh, H., Leonard, J., Walter, M., Ballard, R.: Visually navigating the rms titanic with slam information filters. In: Proceedings of Robotics Science and Systems, Boston. (2005)
11. Castellanos, J., Montiel, J., Neira, J., Tardós, J.: The SPmap: A probablistic framework for simultaneous localization and map building. IEEE Transactions on Robotics and Automation **15**(5) (1999) 948 – 952

3D-6DoF Hierarchical SLAM with 3D vision

D. Marzorati[1], M. Matteucci[2], and D. G. Sorrenti[1]

[1] Universitá di Milano - Bicocca, DISCo, Milano, Italy,
{marzorati,sorrenti}@disco.unimib.it,
[2] Politecnico di Milano, DEI, Milano, Italy
matteucci@elet.polimi.it

Abstract. We motivate and present a SLAM system capable to deal with data from 3D segment-based vision system. These are widespread systems in robotics. Reliable world mapping in large indoor environments is demonstrated by the experimental activity.

1 Introduction

Simultaneous Localization and Mapping (SLAM) deals with the automatic construction of a geometric model of environment [1]. The main issues are related to errors in robot localization and in mapping the world; as a result the world model is affected by geometric inconsistencies. The absence of a reliable SLAM functionality prevents practical use of mobile robotics technology whenever an a priori and up-to-date map of the workspace is not available, i.e. nearly always, as executive drawings (if available) differ from reality, day-to-day usage of space introduces changes such as un-fixed furniture, temporary obstacles, etc.

Many approaches are known in the literature of SLAM systems; some of the most known are Fastslam [2], which decompose the problem in two: robot localization and estimation of the position of the world features and then makes use of a modified particle filter for the estimate of the robot pose and EKF for the map. Graphical SLAM [3] represents the world map as a graph where nodes carry information about the pose of each world feature and the robot; on the edges are the relationship between nodes. Many other approaches base on EKF for building a geometrically consistent map. An interesting technique is Hierarchical SLAM [4], which is based on EKF, Mahalanobis distance, interpretation trees, and hierarchical decomposition of the data structure (which allows a reduction in complexity by limiting the items involved in the most cumbersome SLAM phases. This approach has been used with many sensing systems (sonar, laser range finders, etc.) and also with a 3D vision system like the one we used, but with data obtained by projecting on the floor the 3D data.

In this paper we first shortly recall in Section 2 the specific aspects of our sensed data, and then introduce (Section 3) our 3D-6DoF Hierarchical SLAM work. We then illustrate in Section 4 the experimental activity performed.

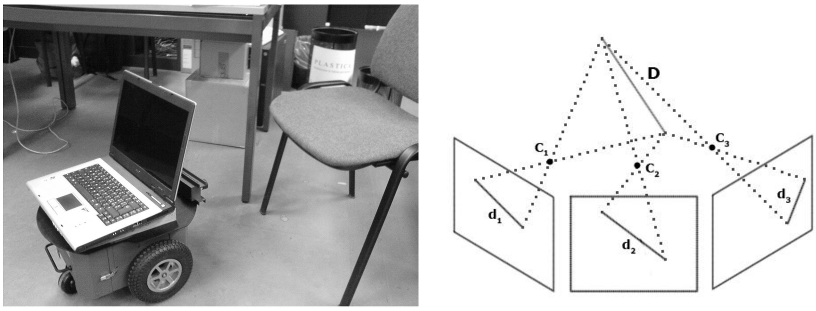
2 Sensor Data

Some robot activity requires a full 3D knowledge of the observed environment features; a few examples are: tables legs or steps which constrain motion; door handles and fire extinguishers help in localization; cleaning has to be performed under tables and chairs; fire extinguishers have to be avoided; books to be moved are on top of tables, etc. In Figure 1a the robot can navigate/clean under the table, but not under the seat. It is therefore relevant to map the real 3D robot workspace, but most of these items are not 3D-perceivable with 2D polar map sensors like LRFs or omnidir. vision. Some existing work dealing with 3D data bases on 3D LRFs [5], but these devices provide just geometric data, which makes difficult other robot tasks. An example could be the semantic classification of places [6], which are required for a real indoor service robot. Even though we are here proposing to use just the geometry provided by 3D vision systems (because we are working with the geometric task of map building), we think that the full richness of vision systems output is necessary for other tasks. On the other hand, it is difficult to put many sensing systems on the same robot, typically because of the cost limitations, like in consumer-level robotics. These considerations are our main motivations for a vision-only 3D-data-based SLAM approach.

The sensing system we use gives out 3D segments, and it is based on the trinocular approach [7]. It deals with segments since the very first processing step. Hence it looks for 2D segments in the image, and then for correspondences between the different images. The last step is the computation of the parameters of the 3D segment, represented by the 3D coordinates of their endpoints. In Figure 1b, $\mathbf{D}$ is the 3D scene segment, $\mathbf{C}_i$ and $\mathbf{d}_i$ are respectively the projection center and the projection of $\mathbf{D}$ on image i. Cameras are calibrated altogether with their covariance matrix so that 3D segment endpoints can be given out altogether with an associated covariance matrix, to represent the measurement uncertainty as a normal probability distribution. Such systems date a long time ago and are quite widespread in the computer vision and robotics communities. Our implementation differs from the original only in the use of the Fast Line Finder [8] in the polygonal approximation phase.

3 3D-6DoF Hierarchical SLAM

In the notation kD-lDoF SLAM, the first item (k) refers to the dimensionality of the data used for building the world model. We use 3D data from the perception system mentioned before. The second item (l) refers to the dimensionality of the observer pose. We model the pose as a full rigid-body transformation, i.e. a 6DoF pose. The whole system is a 3D-6DoF SLAM system, like the one in [5]. On the other hand, the system in [9], which is also based on data from a trinocular system, because of the projecting of data on the floor, is a 2D-3DoF SLAM system. This is a quite common approach in indoor mobile robotics, where the robot is moving on a supposedly flat floor, and it looks reasonable to represent the robot pose with a $< x, y, \vartheta >$ triplet. Our past experience shows that such



(a) (b)

Fig. 1. (a) Cleaning under table/seat requires a 3D understanding of the free space. (b) 3D segment-based reconstruction for a trinocular stereoscopic system.

flatness hypothesis is not realistic and that even with small floor imperfections a complete pose representation, i.e. 6DoF, is of great advantage even in indoor SLAM. In previous work we showed that a higher accuracy can be obtained in pose recovery from images of a scene if a complete 6DoF model of the pose is used [10]. We showed also [11] that a complete, i.e. more realistic, modelling of uncertainties turns also into higher accuracies. In other words, un-modelled uncertainties, i.e. use of deterministic values just due to modelling laziness, bias the estimates. The application scenario in [11] was object localization. We are therefore claiming that a complete (and realistic) modelling of the reality, i.e. 6DoF instead of 3DoF for the robot pose, is of great advantage even in an indoor SLAM scenario.

3.1 Views and Submaps

A view is the set of 3D segments given out in one activation of the perception system. Each segment endpoint is a triple of coordinates, its uncertainty is a 6×6 covariance matrix. Each local map, or submap, is the result of the combination of some views. It contains an estimate of each world feature, i.e. segment, as the result of possibly many observations of it, altogether with the estimate of the robot pose. These data are referred to the same reference frame, which is called "base reference" of the submap. Each submap is therefore a representation of a part of the environment. An important property of submaps is the stochastic independence with respect to other submaps.

At first a new submap is initiated with the output of the perception system. The base reference is put on the current robot pose. The uncertainty on the base reference is null [12]. After some motion the perception system is activated again and a new view generated. The integration of views processing, for details see [13], combines the data in the new view with the data in the submap, e.g. creating a single instance of a world feature from the (possibly) two measures (submap and view) available. Odometry plays a role here because it gives an estimate of the motion. The processing is based on EKF; the state is the union of the vectors of the features and the 6DoF robot pose. Associations are used to update the state. The robot pose, being part of the state, is updated according to the new view data; on long motion sequences this helps limiting to some extent the cumulative odometric errors, see Section 4.

This process is repeated as long as some termination criteria are false. Then the submap is closed and a new submap is initiated at the current robot pose.

3.2 Global Map and Loop Closure

The global map maintains the relationships between submaps as an oriented graph; submaps are the nodes, while the edges represent the spatial relationships between the base references of two submaps, which are therefore considered independent in the features. When closing a submap (i), the last robot pose ($\mathbf{x}$) becomes the base reference of the new submap (j). This pose ($\mathbf{x}_j^i$), with respect to the closed submap base reference, is stored in the graph edge connecting the two submaps.

The loop closure process is a complex activity that involves many steps. The first is obviously *Loop Detection*, i.e. detecting a submap close to the one just closed, that is involved in a loop closure. Once a loop has been detected it is necessary to perform *Data Association* to extract common features. This is obtained by a procedure that seeks an hypothesis H that connects each feature in the first submap to the (possibly) corresponding feature in the second submap. This hypothesis is used to determine both robot and features poses in the submaps. There are different approaches to find H; we use an interpretation tree, as done in [4], exploiting an adapted RANSAC algorithm, which bases on a version of the joint compatibility test [9], adapted to the 3D-6DoF problem. Once data-associations have been found, it is possible to perform *Robot Relocation* and *Local Map Joining*, i.e. to estimate the spatial relationship between the two submaps and thus join the two w.r.t. a common reference frame, therefore creating a single submap. The robot pose is changed w.r.t. the new reference frame as well. *Loop Closure* is the final step in global map building, which allows to reduce the errors in the spatial relationships between the submaps in the loop. Link relaxation in loop closure has been re-formulated as a maxima a-posteriori estimate of all base reference poses under the loop constraint $h(\mathbf{x}) = 0$. To solve this minimization problem we used a Sequential Quadratic Programming approach, similar to what done in [4], which is derived from the Kuth-Tucker equations and has been adapted to the 3D-6DoF problem.

4 Experimental Activity

For the experimental activity we used a mobile robot from Robosoft which computes odometry as a 3DoF pose; this datum reaches a PC via serial line. On the PC we have an Eltec frame grabber capable to grab three 704x558x8 pixels

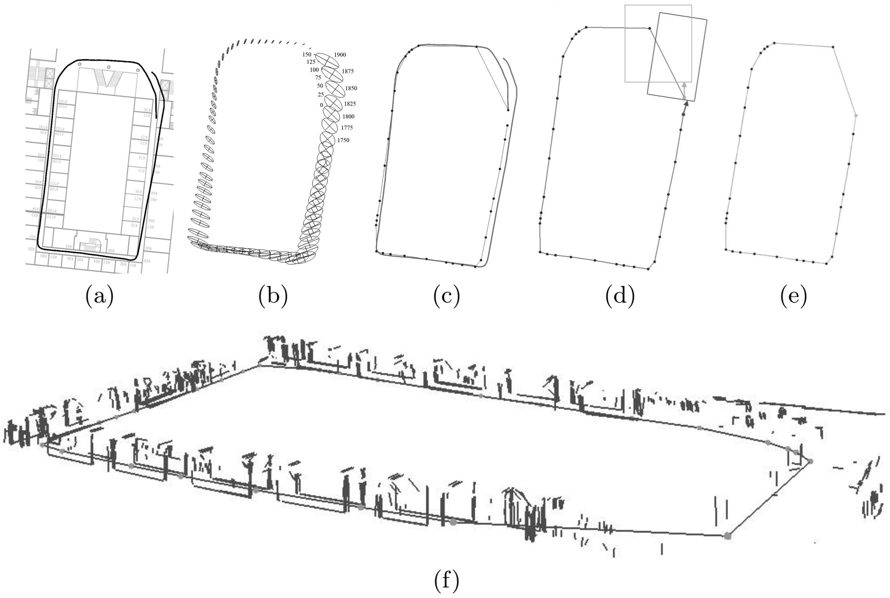


Fig. 2. (a) Odometric travel superimposed to the environment planimetry. (b) Odometric error ellipses ($\pm 3\sigma$), with view_id. (c) Odometric travel (full) superimposed to the base references of the submaps (circles connected by lines); notice the larger accuracy provided by fusion of views. (d) Bounding boxes of last (darker) and first (lighter) submaps. (e) The base references of the submaps after graph relaxation, compare with the base references in (c) or (d). (f) A 3D view of the 6DoF-pose final reconstruction; the solid-circle line is the same as in (e).

images at the same time. Each channel of the frame grabber is connected to a Sony XC75CE camera. Cameras have been calibrated with a standard DLT approach. The robot has been moved, by hand due to a servo-amplifier failure, inside the 4th floor of building U7, Univ. Milano - Bicocca, Milano, Italy. Distances between consecutive robot poses, i.e. views, were about 0.05m. The overall distance travelled has been about 200m. In Figure 2a the odometric travel is shown, altogether with the environment planimetry. The odometric error is modelled as zero mean Gaussian, and the propagation of this error is shown in Figure 2b with the usual 99% ellipses; notice that the actual, i.e. first, poses are in good agreement with the uncertainty propagated up to the last ones. Submap termination is currently set on the cardinality of the features in the submap; the value used in the reported experiment is 50. In Figure 2c the odometric travel is superimposed to the base references of the submaps, i.e. what could be considered as the overall result of the integration of views processing. This figure shows that the integration of views gives a large increase in accuracy, even though this is not enough for obtaining a geometrically consistent map. When a submap is closed loop-detection is activated; in Figure 2d the bounding boxes of the first and last submaps, i.e. the ones for which a loop is detected, are shown. On these two submaps Robot Relocation and Local Map Joining are applied. At the end the two submaps are fused together in a single submap. The geometric consistency is still not attained at this stage. Loop Closure distribute the errors along the whole set of submaps, i.e. relative poses of submaps. The result of such iterative non-linear optimization (graph relaxation) is shown, in terms of base references, in Figure 2e. A 3D view of the 6DoF-pose final reconstruction is presented in Figure 2f.

5 Conclusions

We presented a SLAM system capable to deal with data from 3D segment-based vision system. These are widespread systems in robotics, but SLAM systems basing on them are not common in the literature. Reliable world mapping in large indoor environments is demonstrated by the experimental activity presented.

References

1. J. D. Tardós and J. A. Castellanos, *Mobile robot localization and map building : a multisensor fusion approach*, Kluwer Academic, 1999.
2. S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, “Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association,” *J. of Machine Learning Research*, 2004.
3. J. Folkesson and H. I. Christensen, “Graphical slam - a self-correcting map,” in *Proc. IEEE ICRA*, April 2004, pp. 383–389.
4. C. Estrada, J. Neira, and J. D. Tardós, “Hierarchical slam: real-time accurate mapping of large environments,” *IEEE Trans. on Robotics*, to appear.
5. H. Surmann, A. Nüchter, K. Lingemann, and J. Hertzberg, “6d slam: Preliminary report on closing the loop in six dimensions,” in *5th IFAC Symp. IAV*, June 2004.
6. A. Rottmann, O. Martínez Mozos, C. Stachniss, and W. Burgard, “Place classification of indoor environments with mobile robots using boosting,” in *Proc. of AAAI*, Pittsburgh, PA, USA, 2005.
7. N. Ayache, *Artificial Vision for Mobile Robots*, The MIT Press, 1991.
8. P. Kahn, L. Kitchen, and E. M. Riseman, “A fast line finder for vision-guided robot navigation,” *IEEE Trans. on PAMI*, vol. 12, no. 11, Nov 1990.
9. J. Neira and J. D. Tardós, “Data association in stochastic mapping using the joint compatibility test,” *IEEE Trans. R&A*, vol. 17, no. 6, pp. 890–897, 2001.
10. R. Arpini, V. Caglioti, E. Camnasio, M. Cappelletti, and D. G. Sorrenti, “Visual self-localisation by means of natural landmarks for mobile robot navigation,” *Automazione e Strumentazione*, vol. 43, no. 5, pp. 105 – 115, May 1995, in italian.
11. V. Caglioti, F. Mainardi, M. Pilu, and D. G. Sorrenti, “Improving pose estimation using image, sensor and model uncertainty,” in *Proc. of IEE BMVC Conf.*, 1994, vol. 2, pp. 805–819.
12. J. A. Castellanos, J. Neira, and J. D. Tardós, “Limits to the consistency of ekf-based slam,” in *5th IFAC Symp. IAV*, 2004.
13. D. Marzorati, M. Matteucci, and D. G. Sorrenti, “Multi-criteria data association in 3d-6dof hierarchical slam with 3d segments,” in *Proc. of ASER*, 2006.

6D SLAM with Kurt3D

Andreas Nüchter, Kai Lingemann, Joachim Hertzberg

University of Osnabrück, Institute of Computer Science
Knowledge Based Systems Research Group
Albrechtstr. 28, D-49069 Osnabrück, Germany
{nuechter|lingemann|hertzberg}@informatik.uni-osnabrueck.de

Abstract. 6D SLAM (Simultaneous Localization and Mapping) or 6D Concurrent Localization and Mapping of mobile robots considers six dimensions for the robot pose, namely, the x, y and z coordinates and the roll, yaw and pitch angles. Robot motion and localization on natural surfaces, e.g., when driving with a mobile robot outdoor, must regard these degrees of freedom. 3D (6 DOF) scan matching, combined with a heuristic for closed loop detection and a global relaxation method, results in a highly precise mapping system for outdoor environments. The mobile robot Kurt3D is capable to run the mapping process with its on-board sensors and computers and is used to digitalize different environments. This paper summarizes our previous research.

1 Introduction

Automatic environment sensing and modeling is a fundamental scientific issue in robotics, since the presence of maps is essential for many robot tasks. Manual mapping of environments is a hard and tedious job: Thrun et al. report a time of about one week hard work for creating a map of the museum in Bonn for the robot RHINO [9]. Especially mobile systems with 3D laser scanners that automatically perform multiple steps such as scanning, gaging and autonomous driving have the potential to greatly improve mapping. Many application areas benefit from 3D maps, e.g., industrial automation, architecture, agriculture, the construction or maintenance of tunnels and mines and rescue robotic systems.

The robotic mapping problem is that of acquiring a spatial model of a robot's environment. If the robot poses were known, the local sensor inputs of the robot, i.e., local maps, could be registered into a common coordinate system to create a map. Unfortunately, any mobile robot's self localization suffers from imprecision and therefore the structure of the local maps, e.g., of single scans, needs to be used to create a precise global map. Finally, robot poses in natural outdoor environments necessarily involve yaw, pitch, roll angles and elevation, turning pose estimation as well as scan registration into a problem with six mathematical dimensions.

2 State of the Art

State of the art for metric maps are probabilistic methods, where the robot has probabilistic motion models and uncertain perception models. Through integration of these two distributions with a Bayes filter, e.g., Kalman or particle filter, it is possible to localize the robot. Mapping is often an extension to this estimation problem. Beside the robot pose, positions of landmarks are estimated. Closed loops, i.e., a second encounter of a previously visited area of the environment, play a special role here: Once detected, they enable the algorithms to bound the error by deforming the mapped area to yield a topologically consistent model. However, there is no guarantee for a correct model. Several strategies exist for solving SLAM. Thrun surveys existing techniques, i.e., maximum likelihood estimation, expectation maximization, extended Kalman filter or (sparsely extended) information filter SLAM [10].

SLAM in well-defined, planar indoor environments is considered solved, but 6D SLAM still proposes a challenge, since several strategies become infeasible, e.g., with 6 degrees of freedom the matrices in Kalman or information filter SLAM grow more rapidly and a multi hypothesis approach would require too many particles. Therefore, 3D mapping systems [2–4, 6, 7] often rely on scan matching approaches.

3 Kurt3D

3.1 The 3D laser range finder.

The 3D laser range finder (Fig. 1) [7] is built on the basis of a SICK 2D range finder by extension with a mount and a small servomotor. The 2D laser range finder is attached in the center of rotation to the mount for achieving a controlled pitch motion with a standard servo.

The area of up to 180°(h) × 120°(v) is scanned with different horizontal (181, 361, 721) and vertical (128, 256, 400, 500) resolutions. A plane with 181 data points is scanned in 13 ms by the 2D laser range finder (rotating mirror device). Planes with more data points, e.g., 361, 721, duplicate or quadruplicate this time. Thus a scan with 181 × 256 data points needs 3.4 seconds. Scanning the environment with a mobile robot is done in a stop-scan-go fashion.

3.2 The mobile robot.

Kurt3D (Fig. 1) is a mobile robot with a size of 45 cm (length) × 33 cm (width) × 29 cm (height) and a weight of 22.6 kg. Two 90 W motors are used to power the 6 skid-steered wheels, whereas the front and rear wheels have no tread pattern to enhance rotating. The core of the robot is a Pentium-Centrino-1400 with 768 MB RAM and Linux.

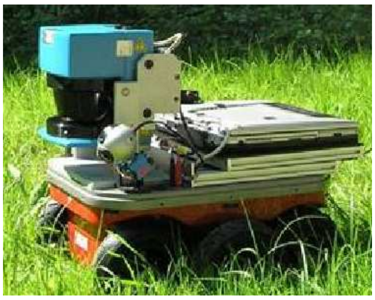


Fig. 1: Kurt3D.

4 6D SLAM

To create a correct and consistent environment map, 3D scans have to be merged into one coordinate system. This process is called registration. If the robot carrying the 3D scanner were localized precisely, the registration could be done directly based on the robot pose. However, due to the imprecise robot sensors, self localization is erroneous, so the geometric structure of overlapping 3D scans has to be considered for registration. As a by-product, successful registration of 3D scans relocalizes the robot in 6D, by providing the transformation to be applied to the robot pose estimation at the recent scan point.

Kurt3D's SLAM algorithm consists of four steps, that are explained in the following subsections.

4.1 Odometry extrapolation

Thh odometry is extrapolated to 6 degrees of freedom using previous registration matrices, i.e., the change of the robot pose $\Delta\mathbf{P}$ given the odometry information $(x_n, z_n, \theta_{y,n})$, $(x_{n+1}, z_{n+1}, \theta_{y,n+1})$ and the registration matrix $\mathbf{R}(\theta_{x,n}, \theta_{y,n}, \theta_{z,n})$ is calculated by solving:

$$\begin{pmatrix} x_{n+1} \\ 0 \\ z_{n+1} \\ 0 \\ \theta_{y,n+1} \\ 0 \end{pmatrix} = \begin{pmatrix} x_n \\ 0 \\ z_n \\ 0 \\ \theta_{y,n} \\ 0 \end{pmatrix} + \left(\begin{array}{c|c} \mathbf{R}(\theta_{x,n}, \theta_{y,n}, \theta_{z,n}) & \mathbf{0} \\ \hline \mathbf{0} & \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix} \end{array}\right) \cdot \underbrace{\begin{pmatrix} \Delta x_{n+1} \\ \Delta y_{n+1} \\ \Delta z_{n+1} \\ \Delta\theta_{x,n+1} \\ \Delta\theta_{y,n+1} \\ \Delta\theta_{z,n+1} \end{pmatrix}}_{\Delta\mathbf{P}} .$$

4.2 Calculating Heuristic Initial Estimations for ICP Scan Matching

For the given two sets M and D of 3D scan points stemming from the 3D scans, our heuristic computes two octrees based on these point clouds. The octrees rigid transformations are applied to the second octree, until the number of overlapping cubes has reached its maximum. The transformations are computed in nested loops. However, the computational complexity is reduced due to the fact that we limit the search space relative to the octree cube size. Details can be found in [4].

4.3 Scan Registration

We use the well-known Iterative Closest Points (ICP) algorithm to calculate a rough approximation of the transformation while the robot is acquiring the 3D scans [1]. The ICP algorithm calculates iteratively the point correspondence. In

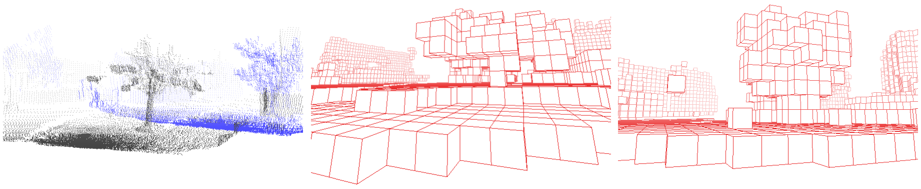


Fig. 2. Left: Two 3D point clouds. Middle: Octree corresponding to the black point cloud. Right: Octree based on the blue points.

each iteration step, the algorithm selects the closest points as correspondences and calculates the transformation $(\mathbf{R}, \mathbf{t})$ for minimizing the equation

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N_m} \sum_{j=1}^{N_d} w_{i,j} \left|\left|\mathbf{m}_i - (\mathbf{R}\mathbf{d}_j + \mathbf{t})\right|\right|^2 , \tag{1}$$

where N_m and N_d, are the number of points in the model set M or data set D, respectively and w_{ji} are the weights for a point match. The weights are assigned as follows: $w_{ji} = 1$, if $\mathbf{m}_i$ is the closest point to $\mathbf{d}_j$ within a close limit, $w_{ji} = 0$ otherwise. The assumption is that in the last iteration step the point correspondences, thus the vector of point pairs, are correct.

4.4 Loop Closing

After matching multiple 3D scans, errors have accumulated and loops would normally not be closed. Our algorithm automatically detects a to-be-closed loop by registering the last acquired 3D scan with earlier acquired scans. Hereby we first create a hypothesis based on the maximum laser range and on the robot pose, so that the algorithm does not need to process all previous scans. Then we use the octree based method presented in section 4.2 to revise the hypothesis. Finally, if a registration is possible, the computed error, i.e., the transformation $(\mathbf{R}, \mathbf{t})$ is distributed over all 3D scans.

4.5 Model Refinement

Based on the idea of Pulli we designed the relaxation method *simultaneous matching* [7]. The first scan is the masterscan and determines the coordinate system. It is fixed. The following three steps register all scans and minimize the global error, after a queue is initialized with the first scan of the closed loop:

1. Pop the first 3D scan from the queue as the current one.
2. If the current scan is not the master scan, then a set of neighbors (set of all scans that overlap with the current scan) is calculated. This set of neighbors forms one point set M. The current scan forms the data point set D and is aligned with the ICP algorithms. One scan overlaps with another iff more than p corresponding point pairs exist. In our implementation, $p = 250$.
3. If the current scan changes its location by applying the transformation (translation or rotation) in step 2, then each single scan of the set of neighbors that is not in the queue is added to the end of the queue. If the queue is empty, terminate; else continue at step 1.

In contrast to Pulli's approach, our method is totally automatic and no interactive pairwise alignment has to be done. Furthermore the point pairs are not fixed [5]. The accumulated alignment error is spread over the whole set of acquired 3D scans. This diffuses the alignment error equally over the set of 3D scans [8].

5 Results and Conclusions

The proposed methods have been tested on various data sets, including test runs at RoboCup Rescue and ELROB. Fig. 3 show two closed loops. 3D animations of the scenes can be found at `http://kos.informatik.uni-osnabrueck.de/download/6Dpre/` and `http://kos.informatik.uni-osnabrueck.de/download/6Doutdoor/`.The loop in the left part of Fig. 3 was closed manually, whereas the right loop was detached automatically.

These large loops require an reliable robot control architecture for driving the robot and efficient 3D data handling and storage methods. In future work we will tackle the emerging topic of map management.

References

1. P. Besl and N. McKay. A method for Registration of 3–D Shapes. *IEEE Transactions on PAMI*, 14(2):239 – 256, February 1992.
2. A. Georgiev and P. K. Allen. Localization Methods for a Mobile Robot in Urban Environments. *IEEE Trans. Robotics and Automation (TRO)*, 20(5), 2004.
3. Martin Magnusson and Tom Duckett. A comparison of 3d registration algorithms for autonomous underground mining vehicles. In *Proc. ECMR*, 2005.
4. A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann. Heuristic-Based Laser Scan Matching for Outdoor 6D SLAM. In *KI, Springer LNAI vol. 3698*, 2005.
5. K. Pulli. Multiview Registration for Large Data Sets. In *Proc. 3DIM*, Oct. 1999.
6. V. Sequeira, K. Ng, E. Wolfart, J. Goncalves, and D. Hogg. Automated 3D reconstruction of interiors with multiple scan–views. In *Proc.of SPIE, Electronic Imaging '99, 11th Annual Symposium*, San Jose, CA, USA, January 1999.
7. H. Surmann, A. Nüchter, and J. Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor en vironments. *Journal Robotics and Autonomous Systems*, 45(3 – 4):181 – 198, December 2003.
8. H. Surmann, A. Nüchter, K. Lingemann, and J. Hertzberg. 6D SLAM A Preliminary Report on Closing the Loop in Six Dimensions. In *Proc. of the 5th IFAC Symp. on Intelligent Autonomous Vehicles (IAV '04)*, Lisabon, Portugal, July 2004.
9. S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
10. S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium.* Morgan Kaufmann, 2002.

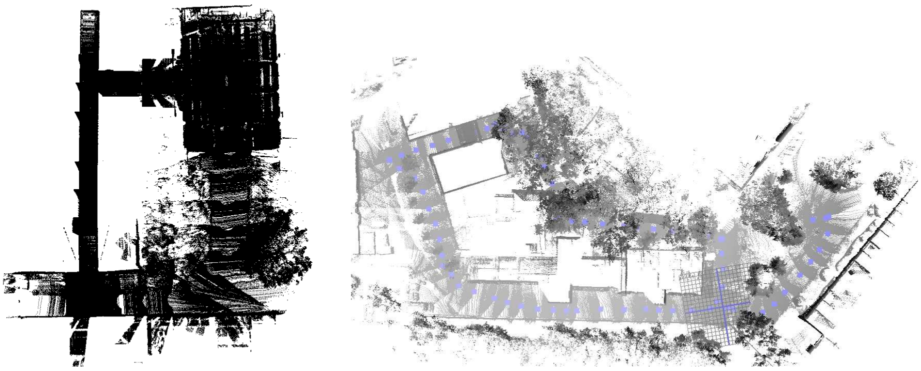


Fig. 3. Two 3D point clouds in top view. Left: Closed loop with 9 million 3D data points. Loop with 7 million points and a path length of over 250 m.