in: Proc. Autonomous Mobile Systems (AMS 2009), Karlsruhe, Germany, pp. 33-40, Informatik Aktuell, Springer 2009

Monocular Obstacle Detection for Real-world Environments

Erik Einhorn[1] and Christof Schroeter[2] and Horst-Michael Gross[2]

[1]MetraLabs GmbH, Germany
[2]Neuroinformatics and Cognitive Robotics Lab
Ilmenau University of Technology, Germany

Abstract. In this paper, we present a feature based approach for monocular scene reconstruction based on extended Kalman filters (EKF). Our method processes a sequence of images taken by a single camera mounted in front of a mobile robot. Using different techniques we are able to produce a precise reconstruction that is free from outliers and therefore can be used for reliable obstacle detection and avoidance. In real-world field-tests we show that the presented approach is able to detect obstacles that can not be seen by other sensors, such as laser-range-finders. Furthermore, we show that visual obstacle detection combined with a laser range finder can increase the detection rate of obstacles considerably allowing the autonomous use of mobile robots in complex public and home environments.

1 Introduction

For nearly ten years we have been involved with the development of an interactive mobile shopping assistant for everyday use in public environments, such as shopping centers or home improvement stores. Such a shopping companion autonomously contacts potential customers, intuitively interacts with them, and adequately offers its services, including autonomously guiding customers to the locations of desired goods [1]. Currently, we are developing an interactive assistant that can be used in home environments as companion for people with mild cognitive imparments (MCI) living at home alone. However, both public emvironments, like home improvement stores, as well as home environments like kitchens or living rooms, contain a large variety of different obstacles that must be detected by an autonomous robot.

For obstacle detection our robot is equipped with an array of 24 sonar sensors at the bottom and a laser range finder SICK S300 mounted in front direction at a height of 0,35 meter. Using these sensors, many obstacles can be reliably detected. However, during the field trials it became apparent that many obstacles are very difficult to recognize. Some obstacles are mainly located above the plane that is covered by the laser range finder. Also small obstacles are difficult to reveal since they lie below the laser range finder and can hardly be seen by the sonar sensors due to their diffuse characteristics and low precision. Therefore, it turned out to be necessary to use additional methods for robust and reliable obstacle detection. Vision-based approaches are suitable for this purpose since they provide a large field of view and supply a large amount of information about the structure of the local surroundings.

Recently, time-of-flight cameras have been used successfully for obstacle detection [2]. Similar to laser range finders, these cameras emit short light pulses and measure the time taken until the reflected light reaches the camera again. Another alternative is to use stereo vision for obstacle detection as described in [3] and many others. However, a stereo camera is less compact than a single camera. Furthermore, a monocular approach that uses one camera only is more interesting from a scientific point of view.

In [4] monocular approach for depth estimation and obstacle detection is presented. Information about scene depth is drawn from the scaling factor of image regions, which is determined using region tracking. While this approach may work well in outdoor scenes, where the objects near the focus of expansion are separated from the background by large depth discontinuities, it will fail in cluttered indoor environments like home improvement stores or home environments. In [5] we propose an early version of a feature-based approach for monocular scene reconstruction. This shape-from-motion approach uses extended Kalman filters (EKF) to reconstruct the 3D position of the image features in real-time in order to identify potential obstacles in the reconstructed scene. Davison et al. [6,7] use a similar approach and have done a lot of research in this area. They propose a full covariance SLAM algorithm for recovering the 3D trajectory of a monocular camera. Both, the camera position and the 3D positions of tracked image features or landmarks are estimated by a single EKF. Another visual SLAM approach was developed by Eade and Drummond [8]. Their graph-based algorithm partitions the landmark observations into nodes of a graph to minimize statistical inconsistency in the filter estimates [9].

However, Eade's and Drummond's "Visual SLAM" as well as Davison's "MonoSLAM" are both mainly focusing on the estimation of the camera motion, while a precise reconstruction of the scenery is less important. As we want to use the reconstructed scene for obstacle detection, our priorities are vice versa. We are primarily interested in a precise and dense reconstruction of the scene and do not focus on the correct camera movement, since the distance of the objects relative to the camera and the robot respectively is sufficient for obstacle avoidance and local mp building. Actually, we are using the robot's odometry to obtain information on the camera movement. In contrast to Eades and Davison who generally move their camera sidewards in their examples, our camera is mounted in front of the mobile robot and, therefore, moves along its optical axis. Compared to lateral motion, this forward motion leads to higher uncertainties in the depth estimates due to a smaller parallax. This fact was also proven by Matthies and Kanade [10] in a sensitivity analysis.

The main contribution of this paper is a monocular feature-based approach for scene reconstruction that combines a number of different techniques that are known from research areas like visual SLAM or stereo vision to achieve a robust algorithm for reliable obstacle detection that must fulfill the following requirements:

1. A dense reconstruction to reduce the risk of missing or ignoring an obstacle
2. The positions of obstacles that appear in the field of view should be correctly estimated as early as possible to allow an early reaction in motion control
3. Outliers must be suppressed to avoid false positive detections that result in inadequate path planning or not necessary avoidance movements

The presented algorithm is based on our previous work [5] and was improved by several extensions. In the next sections, we describe our approach in detail and show how it can be used for visual obstacle detection. In section 4 we present some experimental results and conclude with an outlook for future work.

2 Monocular Scene Reconstruction

As stated before, we use a single calibrated camera that is mounted in front of the robot. During the robot's locomotion, the camera is capturing a sequence of images that are rectified immediately according to the intrinsic camera parameters. Thus, different two-dimensional views of a scene are obtained and can be used for the scene reconstruction. In these images distinctive image points (image features) are detected. For performance reasons we use the “FAST” corner detector [11] since SIFT or SURF features still require too much computation time. The selected features are then tracked in subsequent frames while recovering their 3D positions.

Davison et al. [6,7] use a single EKF for full covariance SLAM that is able to handle up to 100 features. As we require a denser reconstruction of the scene for obstacle detection, we have to cope with a large number of features which cannot be handled by such an approach in real-time. Therefore, we decouple the the EKF and use one EKF per feature to recover the structure of the scene similar to [12]. Each feature i is associated with a state vector $\mathbf{y}_i$ that represents the 3D position of the feature and a corresponding covariance matrix $\mathbf{\Sigma}_i$.

2.1 State Representation

Different parametrizations for the 3D positions of the features have been proposed in literature. The most compact representation is the XYZ-representation where the position of each feature is parameterized by its Euclidean coordinates in 3-space. Davison et al. [7] have shown that this representation has several disadvantages since the position uncertainties for distant features are not well represented by a Gaussian distribution. Instead, they propose an inverse depth representation, where the 3D position of each feature i can be described by the vector $\mathbf{y_i} = (\mathbf{c}_i \, , \, \theta_i \, , \, \varphi_i \, , \, \lambda_i)^\top$, where $\mathbf{c}_i \in \mathbb{R}^3$ is the optical center of the camera from which the feature i was first observed, and θ_i, ϕ_i is the azimuth and elevation of the unit ray that points from $\mathbf{c}_i$ to the 3D point of the feature. This ray is given by $\mathbf{m}\left(\theta_i, \phi_i\right) = \left(\cos\theta_i \cos\phi_i \, , \, \cos\theta_i \sin\phi_i \, , \, -\sin\theta_i\right)^\top$. The last element λ_i of the state vector denotes the inverse of the features depth $d_i = \lambda_i^{-1}$ along the ray.

2.2 Feature Tracking

While the robot is moving, the image features are tracked in subsequent frames. In [5] we used a feature matching approach that finds correspondences between homologue features in subsequent frames based on a bipartite graph matching. While that approach is suitable for SIFT or SURF features it has some shortcomings with less complex feature descriptors like image patches. Here, we use a guided active search for tracking the features through the image sequence. As descriptor we utilize a 16×16 pixel image patch around each feature. First, the image position $\mathbf{x}_i^-$ of each feature is predicted by projecting the current estimate of its estimated 3D position $\mathbf{y}_i$ onto the image plane using $\tilde{\mathbf{x}}_i^- = h\left(\mathbf{y}_i, \mathbf{P}\right)$ with the measurement function[1]:

$$h\left(\mathbf{y}_i, \mathbf{P}\right) = \mathbf{P}\left(\lambda \tilde{\mathbf{c_i}} + \begin{pmatrix} \mathbf{m}\left(\theta_i, \phi_i\right) \\ 0 \end{pmatrix}\right) . \tag{1}$$

Here $\mathbf{P} = \!\mathbf{K}\mathbf{R}\left[\mathbf{I} \,|\, -\mathbf{c}\right]$ is the projection matrix containing the current orientation $\mathbf{R}$, the current position $\mathbf{c}$ and the intrinsic calibration matrix $\mathbf{K}$ of the camera, which captured the current image (see [13] for details). The current camera pose is obtained from the robot’s odometry data.

[1] For better differentiation we notate homogeneous vectors as $\tilde{\mathbf{x}}$ and Euclidean vectors as $\mathbf{x}$, where $\tilde{\mathbf{x}} = (\mathbf{x}, 1)^\top \cdot s \, , \, s \in \mathbb{R}$

For each feature i, the corresponding image point is searched in the current image around the predicted image position $\mathbf{x}_i^-$ by computing the sum of absolute differences (SAD) with the image patch that is stored as descriptor of the feature. The image point that yields the lowest SAD is chosen. To achieve sub-pixel precision we fit a 2D parabola into the computed SAD error surface around the chosen image point and use the coordinate of the apex as position of the corresponding image point. The search is restricted to an elliptical region that is defined by convariance matrix of the innovation that is computed in the EKF.

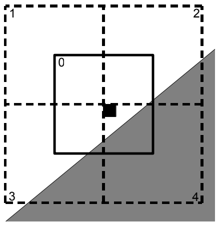

Fig. 1. The correlation window is split into 5 sub-windows.

One major problem of patch-based approaches for feature matching are occlusions near object edges where the patch covers two different objects with large depth discontinuities. During the matching, this leads to a decision conflict since the part of the patch that belongs to the background object moves in a different way than the foreground object. As a result, the reconstructed 3D points along object borders are blurred in different depths. For stereo matching different adaptive window approaches have been proposed to reduce this problem.

We apply a variation of the multiple window approach presented in [14] and [15]. Instead of using a single 16×16 pixel correlation window, the window is split into five sub-windows as shown in Figure 1. The SADs are computed for each sub-window C_i. The final correlation value C is formed by adding the correlation value C_o of the central sub-window and the values of the two best surrounding correlation windows C_b and C_s. This measure of similarity performs better near object boundaries since at least two sub-windows are located on a single object in most cases. Depending on the dominant image structure the correspondence is either attached to foreground or the background object reducing the blur along the reconstructed object borders. Using the SSE2 processor instruction `PSADBW` the correlation values can be computed efficiently and splitting the window into 5 sub-windows results in very little computational overhead compared to a single correlation window. This performance improvement is a major reason for choosing the SAD as measure of similarity. Besides the correlation value, we compute an occlusion score C_{occ} by adding the correlation values of the two worst matching surrounding sub-windows. Both the correlation value C and the occlusion score C_{occ} are normalized by the number of pixels in the used sub-windows.

2.3 Descriptor Update

Davison el al. [7] also use the image patch around the feature as descriptor. While they capture this descriptor only once when the feature is first observed, we used a contrary philosophy in [5], where we update the descriptor every time the feature is tracked in a new image. Both variants have pros and cons. If the descriptor is never updated, the feature cannot be tracked over long distances since the appearance changes too much due to affine and perspective deformations, especially when using a forward moving camera or robot. If, on the other hand, the descriptor is updated every frame, tracking errors might be accumulated over several frames, and the descriptor might move along the edges of object boundaries and does not represent a single fixed feature. This usually occurs near occlusions and leads to incorrect estimates.

Therefore, we use the aforementioned occlusion score C_{occ} to determine whether updating the descriptor is reasonable or not. If the normalized occlusion score C_{occ} lies below a certain the threshold descriptor is updated using the corresponding patch in the current image, otherwise the descriptor remains unchanged. Using this technique, most features can be tracked over long distances while the projective deformations are compensated by permanent descriptor updates. Feature descriptors near occlusions are not updated to allow stable tracking along object boundaries.

After the features are tracked and the camera pose is refined, the 3D positions of the features will be updated using the usual EKF update equations leading to a more precise reconstruction of the scenery.

2.4 Feature Initialization

Lost features that left the camera's field of view or that cannot be tracked in the previous step are replaced be new features. Different methods for initializing the state of new features have been proposed in related literature. In [5] we have shown how to use a multi-baseline stereo approach for initializing new features. The approach uses the images that were captured *before* the feature was first detected and searches along the epipolar line for corresponding image regions by computing the SAD. By accumulating the SAD error over multiple images a reliable initial inverse depth estimate is obtained. Additionally, we treat the SAD error along the epipolar line as probability distribution and fit a Gaussian distribution near the minimum in order to obtain a variance of the initial estimate that is used for initializing the error covariance matrix $\mathbf{\Sigma}_i$.

3 Obstacle Detection

For obstacle detection, we perform the described monocular scene reconstruction for 200-300 salient features of the scene simultaneously. Afterwards, the reconstructed features have to undergo some post-processing where outliers and unreliable estimates are removed. From all reconstructed features, we only use those that meet the following criteria:

- the estimated height must be above 0.1m; obstacles below this threshold cannot be detected safely
- the variance of the estimated inverse depth taken from the error covariance matrix must lie below a threshold of 0.005
- the distance to the camera must have been smaller than 3m when the feature was observed for the last time

The last criterion mainly removes virtual features that arise where the boundaries of foreground and background objects intersect in the image. These features do not correspond to a single 3D point in the scene and cannot be estimated properly.

The features that pass the above filters may still contain a few outliers. Therefore, we examine the neighborhood of each feature. Features that contain less than 4 neighbors within a surrounding sphere with a radius of 0.3m are regarded as outliers and will be rejected. The remaining features are inserted into an occupancy map by projecting them on the xy-plane. This occupancy map is merged with a laser map by choosing the highest probability for each cell in both maps. Finally, the merged map is used for both local path planning and obstacle avoidance.

4 Results

Figure 2 shows such a map where laser and visual information is merged. The occupancy map that is created using the laser range finder is colored in blue where the different shades of blue correspond to the probability that a cell is occupied. The position of the features that were reconstructed using visual information and the approach presented in this paper are colored in red. In the map, a total number of about 8,200

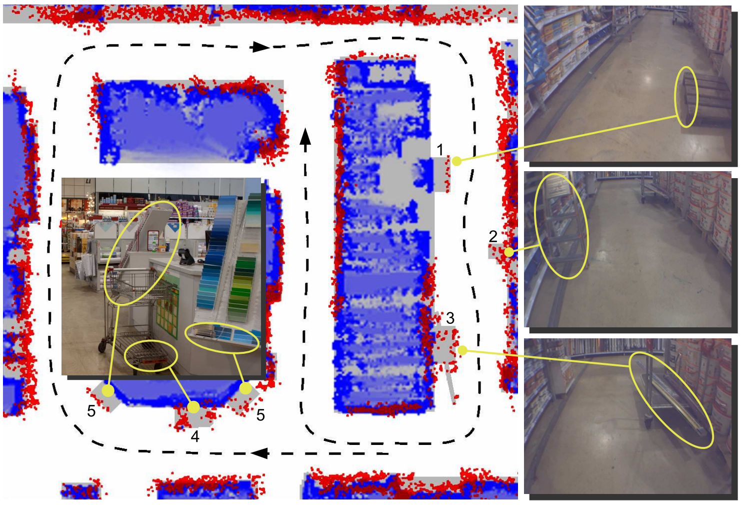

Fig. 2. Map created by combining visual information (red dots) and laser range finder (blue). The robot's trajectory and moving direction is denoted by the dashed line. The ground truth is highlighted in gray. The visual map consists of about 8,200 reconstructed points. Obstacles detected using vision only are labled using numbers. The images on the right show the obstacles as seen by the front camera. The image on the left was taken using a handheld 8 megapixel camera.

visual features is shown. While creating the map a total number of 15,400 points was reconstructed, where 6,000 features where filterd due to a bad variance, 1,000 features were classified as belonging to the ground and 100 where detected as outliers. For image acquisition a 1/4" CCD fire-wire camera is installed on the robot that is mounted at a height of 1.15m and tilted by 35° towards the ground.

For better evaluation and for visualization purposes a ground truth map was created and is highlighted in gray in the background of Figure 2. For building the ground truth, we took images of the scene using a hand held Canon EOS 350D 8.0 megapixel camera and used a bundle adjustment tool[2] for creating a precise reconstruction of the scene which finally was edited and labeled manually.

The map covers an area of 14m×12m within a home improvement store where our tests were conducted. This test area contains typical obstacles that we identified as problematic during the field test since they cannot be detected by the laser range finder due to their reflection properties, their form, or too low height. Some of these obstacles are numbered from 1 to 5 in Figure 2. In detail these obstacles are: 1. an empty Europallet with a height of 11cm, 2. a ladder, 3. a low shopping cart with goods that jut out at both ends, 4. a high shopping cart, and 5. shelves that extend into the scene.

All of these obstacles cannot be seen by the laser range finder and, therefore, might result in collisions. However, using our visual approach these obstacle can be detected safely. In Table 1 we try to quantify this result. For each obstacle, we have manually labeled those parts of the outline that are relevant for navigation and obstacle avoidance during the above test run using the ground truth map. The statistics in Table 1 show

[2] Bundler: `http://phototour.cs.washington.edu/bundler/`

the percentage of the relevant obstacle boundaries that were detected by our visual approach, the laser range finder and a combination of vision and laser. These results show that major parts of the above mentioned obstacles can be detected. Furthermore, it can be seen that the detection rate for all relevant objects in the scene can be increased significantly by 20% compared to obstacle detection using a laser range finder only.

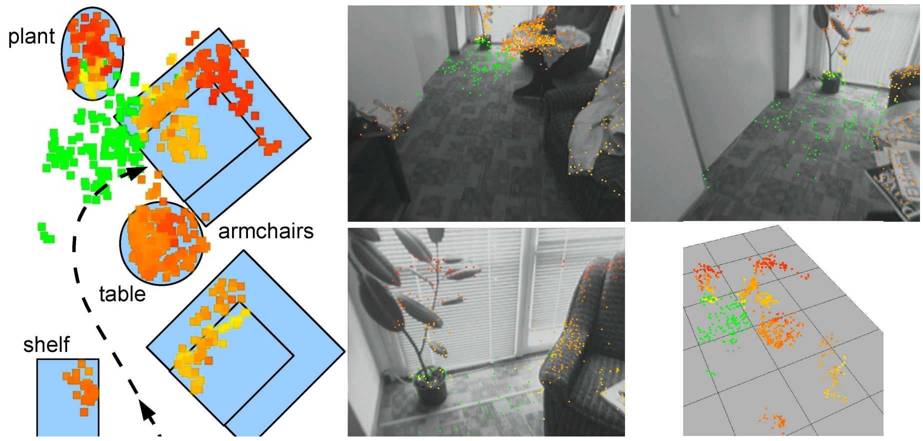

Fig. 3. left: top view of our test area, the reconstructed features are shown as colored dots, where the color indicates the estimated height of each feature (green: < 0.10m, yellow-red: 0.10m-1.15m), **right:** images of the scene superimposed by highlighted features, **lower right:** synthetic 3D view of the estimated features.

Additional tests were carried out in a special test area of our lab that contains typical elements of a living room as well as a floor with repetitive texture. Figure 3 shows a top view of this test area. The reconstructed features are shown as colored dots, where the color indicates the estimated height of each feature. All obstacles that were covered by the camera are detected robustly, while features on the floor are estimated correctly and classified as free and passable. The images on the right of Figure 3 show three images of the scene taken by the front camera as well as a synthetic three-dimensional view of the reconstructed features.

obstacle	visual	laser	visual+laser
1	63%	-	63%
2	71%	-	71%
3	71%	-	71%
4	68%	10%	68%
5	82%	-	82%
others	85%	78%	96%
total	**83%**	**72%**	**93%**

Table 1. Percentage of obstacle boundaries that can be detected using the presented visual approach, a laser range finder and a combination of both for the 5 labeled obstacles and the rest of the scene shown in Figure 2.

All tests were conducted on an Intel Core 2 Duo, 2 GHz CPU. In spite of utilizing one core only we are able to process up to 30 frames per second while reconstructing 200-300 features simultaneously. Depending on the robot's driving speed, we only need to process 10-15 frames per second leaving enough CPU resources for other applications like map building, navigational tasks, user tracking and human-machine interaction.

5 Conclusion and Future Work

In this paper, we have presented an algorithm for monocular scene reconstruction and shape from motion. We have described some improvements that make the reconstruction more reliable and help to reduce outliers. These techniques allow the approach to be used for robust real-time obstacle detection. In realistic field tests, we have shown that some obstacles that are not visible to sensors like laser range finders can be safely detected by the vision based approach. Furthermore, we were able to show that visual obstacle detection combined with a laser range finder can increase the detection rate of obstacles considerably. During the next months we will carry out long-term tests to evaluate how much the number of collisions or near-collisions can be decreased during the daily usage of the robots.

Currently, we are developing a method to estimate the position of moving objects. However, since the position of moving objects can be reconstructed up to a scaling factor only, we will focus on obstacles that reach to the ground. At the moment, features along moving objects are rejected while feature tracking and filtered after the reconstruction due to their high variance in the position estimate.

References

1. H.-M. Gross, H.-J. Böhme, Ch. Schröter, St. Müller, A. König, Ch. Martin, M. Merten, and A. Bley. ShopBot: Progress in Developing an Interactive Mobile Shopping Assistant for Everyday Use. In *SMC*, pages 3471–3478, Singapore, 2008.
2. T. Schamm, S. Vacek, J. Schröder, J.M. Zöllner, and R. Dillmann. Obstacle detection with a Photonic Mixing Device-camera in autonomous vehicles. *International Journal of Intelligent Systems Technologies and Applications*, 5:315–324, Nov. 2008.
3. P. Foggia, J.M. Jolion, A. Limongiello, and M. Vento. Stereo Vision for Obstacle Detection: A Grap-Based Approach. *LNCS GbRPR*, 4538:37–48, 2007.
4. A. Wedel, U. Franke, J. Klappstein, T. Brox, and D. Cremers. Realtime Depth Estimation and Obstacle Detection from Monocular Video. *DAGM*, pages 475–484, 2006.
5. E. Einhorn, Ch. Schröter, H.-J. Böhme, and H.-M. Gross. A Hybrid Kalman Filter Based Algorithm for Real-time Visual Obstacle Detection. In *ECMR*, pages 156–161, 2007.
6. A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Trans. on PAMI*, 29(6):1052–1067, 2007.
7. J. Civera, A.J. Davison, and J. Montiel. Inverse Depth Parametrization for Monocular SLAM. *IEEE Trans. on Robotics*, 24(5):932–945, Oct. 2008.
8. E. Eade and T. Drummond. Monocular SLAM as a Graph of Coalesced Observations. In *IEEE Int. Conference on Computer Vision, ICCV*, pages 1–8, 2007.
9. E. Eade and T. Drummond. Unified Loop Closing and Recovery for Real Time Monocular SLAM. In *Proc. of the British Machine Vision Conference, BMVC*, 2008.
10. L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–238, 1989.
11. E. Rosten and T. T. Drummond. Machine learning for high-speed corner detection. In *Proc. of the European Conference on Computer Vision*, volume 1, pages 430–443, 2006.
12. Y. Yu, K. Wong, and M. Chang. A Fast Recursive 3D Model Reconstruction Algorithm for Multimedia Applications. In *ICPR*, volume 2, pages 241–244, 2004.
13. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0-521-54051-8, second edition, 2006.
14. H. Hirschmüller, P. Innocent, and J. Garibaldi. Real-Time Correlation-Based Stereo Vision with Reduced Border Errors. *International Journal of Computer Vision*, 47:229–246, 2002.
15. W. van der Mark and D.M. Gavrila. Real-time dense stereo for intelligent vehicles. In *IEEE Transactions on Intelligent Transportation Systems*, 2006.