Histogram of Oriented Cameras - A New Descriptor for Visual SLAM in Dynamic Environments

Katrin Pirker
kpirker@icg.tugraz.at
Matthias Rüther
ruether@icg.tugraz.at
Horst Bischof
bischof@icg.tugraz.at

Institute for Computer Graphics and Vision
Graz University of Technology
Graz, Austria

Abstract

Simultaneous localization and mapping (SLAM) is a basic prerequisite in autonomous mobile robotics. Most existing visual SLAM approaches either assume a static environment, or simply 'forget' old parts of the map to cope with map size constraints and scene dynamics. We present a novel map representation for sparse visual features. A new 3D point descriptor called Histogram of Oriented Cameras (HOC) encodes anisotropic spatial visibility information and the importance of each three-dimensional landmark. Each feature holds and updates a histogram of the poses of observing cameras. It is hereby able to estimate its probability of occlusion and importance for localization from a given viewpoint. In a series of simulated and real-world experiments we prove that the proposed descriptor allows to cope with dynamic changes in the map, improves localization accuracy and enables reasonable control of the map size.

1 Introduction

Simultaneous Localization and Mapping (SLAM) is the problem of position estimation in a previously unknown environment, and simultaneously and incrementally building an environmental map. It is an essential prerequisite for many high level applications such as autonomous navigation, path planning or obstacle avoidance (e.g. [26, 31]). Currently there exists an abundance of proposed solutions, using a variety of different sensors in indoor, outdoor, underwater or airspace environments.

Most existing solutions assume a static environment containing only stationary objects. The map is either continuously updated, or constructed once and used for localization afterwards. Rapid environmental changes like moving persons are usually filtered out, while changes with longer duration distort the map. SLAM approaches, which are based on dense range readings or higher level object representations, might be able to detect and handle such changes [6, 29], but approaches based on sparse local features can not. The key problem thereby is how to handle map features which should be visible from a certain viewpoint, but are actually not observed. Most existing SLAM approaches simply add all incoming sensor

BMVC 2010 doi:10.5244/C.24.76

readings over time but do not reject existing features anymore. This results in ever growing maps and may also lead to data association problems.

In this work we address the SLAM problem based on local features within a Structure from Motion (SFM) framework. Our sensor is a stereo camera, producing a set of local feature points at each time step, and associated local descriptors. We address the following problems:

- How to handle short- and long-term environmental changes.
- How to balance the map size.
- How to improve data association in a dynamic environment.

We developed a new 3D descriptor to incorporate visibility information and feature importance through a three-dimensional histogram centered around each landmark. Each descriptor bin tracks how often its associated landmark has been observed from a specific location. Hence, short- and long-term dynamics do not affect localization and the constructed map implicitly adapts to dynamic changes during mapping.

2 Related Work

While SLAM approaches based on sparse local features exist for a number of sensor modalities, we mainly concentrate on vision systems, subdivided into two categories: those following a probabilistic approach by recursively updating the probability of feature location and camera pose, and geometric approaches incorporating SFM techniques to refine map geometry.

Davison *et al.* [4] and Eade and Drummond [7] developed monocular SLAM systems using an extended Kalman filter (EKF) and a particle filter respectively. Using EKF, the size of the filter matrix is cubic in the number of features and filter updates become very costly. The particle filter approach requires lots of particles to track the robot pose. Recently, authors proposed to split the mapping procedure to local submaps, followed by a routine to merge them (e.g. [24, 25]). Chli *et al.* [2] propose a tree-like hierarchy where 3D point features are grouped into clusters from coarse (independent) to fine (all grouped together) to speed up monocular Kalman Filter SLAM. In order to reduce the size of the EKF filter matrix Gee *et al.* [10] fit planes to pointcloud data.

Klein and Murray [16, 17] use sparse bundle adjustment over selected keyframes to refine camera pose and map structure. They successfully mapped a small office environment with a single hand-held camera only. They also fused their Parallel Tracking and Mapping framework [15] with a bundle adjustment approach proposed by Sibley *et al.* [27] using a relative representation of camera poses and 3D points to reduce computational effort. Similarly, Konolige and Agrawal [18] developed FrameSLAM, a stereo mapping approach. They reduced computational effort of bundle adjustment on large feature maps through a nonlinear reduction of frames and image measurements.

All mentioned approaches assume a static environment and perform mapping in a single run only. Research focusing on dynamic environments is sparse and can be categorized in two fields: those focusing on short-term changes only, such as moving people or cars within an otherwise static environment, and others concentrating on less frequent changes within lifelong operations. Wang *et al.* [30] proposed to maintain a stationary and a dynamic occupancy grid map, constructed out of laser data. Lidoris *et al.* [20] combined a Rao-Blackwellized particle filter for robot pose estimation with a person tracker for moving object detection. Other authors try to filter out false measurements from 2D laser data before incorporating them into an occupancy grid map [6], or use map matching between the current scan and the already generated map for moving object detection [29]. There also exist purely statistical approaches to retrieve the most likely map within highly dynamic scenes as proposed by Hähnel *et al.* [11].

In dynamic visual map building the main research focus lies in rejecting independently moving objects. An approach using stereo-cameras to estimate the robot position, the static map and the trajectory of moving objects is proposed by Sola *et al.* [28]. They use a Kalman filter for each moving object to keep track of its pose within the map. Migliore *et al.* [22] use two separate EKFs for the static and dynamic parts in the scene to keep the state vector (pose and landmarks) small, and to track the dynamic scene elements. They propose to use uncertain projective geometry to detect dynamic elements. Results are provided only within a small office scene using a few features.

Little work has addressed the problem of long-term mapping. Dissanayake *et al.* [5] reduced the computational effort within an EKF-based SLAM algorithm by erasing laser range landmarks with low information. The information content is estimated through the diagonal elements of the covariance matrix. They showed that localization errors within the reduced map are relatively small compared to standard EKF approaches. Biber and Duckett [1] create occupancy maps from laser data at different timescales to incorporate new elements while preserving the old and stable ones. They evaluated their method based on localization accuracy over several weeks. Recently, Hochdorfer and Schlegel [14] addressed the problem of ever growing number of landmarks within a feature based map, especially in life-long operations. To avoid extensive growing of the EKF state-vector, they limit the number of allowed landmarks in a two stage process: First, k-means clustering combines points which are observed from neighboring robot poses. Second, landmarks with the lowest localization benefit within each cluster, estimated out of their covariances, are removed. Similarly, Konolige and Bowman [19] adapted FrameSLAM [18] to update a given map in case of new or removed features and to recover from localization failure. They first build a connectivity graph between keyframes, based on the number of successful SIFT features, and delete those keyframes with a very high SIFT matching percentage to the neighbors. They evaluated their system in a dynamic indoor environment of about $50 \times 50m^2$, including moving people and various lighting conditions. They successfully managed to update a map after removed and added furniture and kept the number of keyframes relatively small.

To summarize, all vision based methods use either object detection and tracking to separate moving objects from the static map, or make use of spatial clustering in combination with heuristics to discard weak features. They require prior knowledge of the scene and the type of moving objects to track them adequately. In contrast, our approach encodes feature visibility during mapping and constructed map automatically adapts within a changing environment.

3 Feature Descriptor

To implicitly handle the ambiguity between scene dynamics and occlusion, we propose to add spatial visibility information to local map features. To encode the visibility and importance of each three-dimensional landmark in a map, we develop the Histogram of Oriented Cameras (HOC) descriptor.

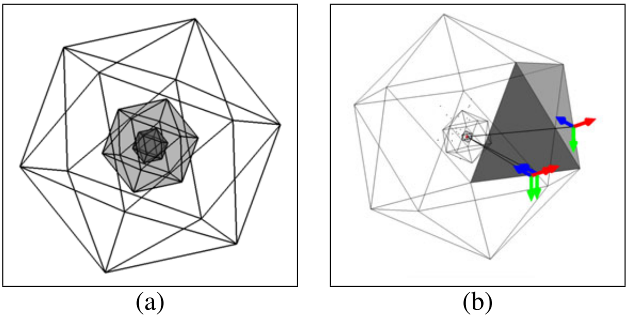


Figure 1: Proposed HOC-descriptor. (a) An uninitialized HOC-descriptor around a feature point consisting of three depth layers. (b) Update of the descriptor with three sensor positions. Darker colors indicate a higher weighted bin.

Each map feature holds information of its location, a descriptor for data association, and a histogram which keeps track of how often the feature has been observed from a specific location. The histogram partitions the space around a feature by a set of k concentric spheres with radii $r_1,...,r_k$. Each sphere S_i is approximated by an icosahedron consisting of m faces f_{ij}, $j = 1,...,m$ (see Figure 1(a)). The discrete polyhedral approximation allows us to partition the sphere surface in triangles of equal size and results in a fast bin search given a camera position. A single histogram bin corresponds to the volume V_{ij} of a triangular pyramid frustum between two consecutive radii r_i, r_{i+1}, limited by two corresponding faces $f_{ij}, f_{(i+1)j}$. A logarithmic spacing of the radii allows us to cover a large volume around each landmark, assuming that spatial partitioning is more important for the closer features.

Given a camera pose and a HOC descriptor, we determine the associated histogram bin V_{ij} in the following way: First, we determine the corresponding radius interval by calculating the Euclidean distance d between feature and camera center. Second, intersection between the icosahedron and the line from feature to camera center returns the valid face f_j. We organized the circumcenters of each triangle in a kd-tree. After projecting the camera center onto the sphere a nearest-neighbor search in the tree returns the corresponding face. The kd-tree is hereby identical for all HOC-descriptors, and needs to be stored only once.

Each histogram consequently consists of km bins. Each bin holds an integer n_{ij}, corresponding to the number of observations from sensors resting in V_{ij}. It is important to note that n_{ij} is increased in case of a positive observation, and decreased if the sensor should observe the feature, but did not produce a positive match.

From n_{ij}, an importance weight p_{ij} is calculated, according to a Sigmoid function:

$$p(n_{ij}) = \frac{1}{1+e^{-\lambda n_{ij}}}, \tag{1}$$

where λ is a user defined scalar ranging between 0.3 (low dynamic scene) and 0.9 (high dynamic scene). The higher p, the more probable it is to observe this feature with a sensor resting in bin volume V_{ij}. The bin value n is clamped such that $0.05 < p(n) < 0.95$. Figure 1(b) shows the update procedure of a HOC-descriptor given three camera positions, where darker gray values indicate a higher weight.

This descriptor allows us to add the following information to the map:

- spatially constrained visibility by increasing n_{ij},

- probable occlusion/dynamics by decreasing n_{ij},
- probably vanished features by looking at the histogram maximum.

Additionally, different feature descriptors may be added to the histogram bins to model view-dependent appearance.
How this descriptor can be embedded in a SLAM framework, especially in dynamic environments, is addressed in the following section.

4 Simultaneous Localization and Mapping

The concept of our descriptor is general, we demonstrate its applicability to visual SLAM. Our SLAM algorithm is closely related to the work of Klein and Murray [16]. A hand-held stereo-rig is used to estimate the sensor pose whilst building a sparse map of three-dimensional feature points. In the following, the algorithm is described. Later, the HOC-descriptor is added to the method.

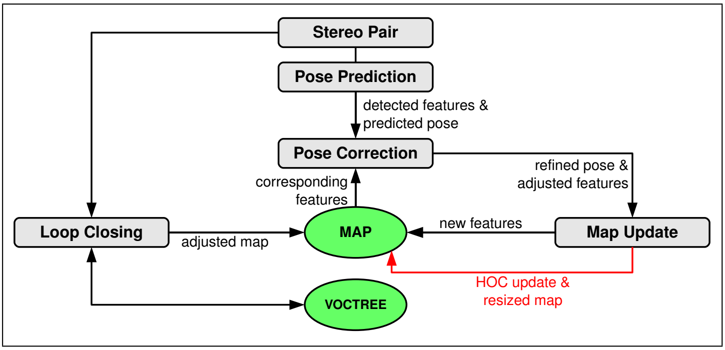


Figure 2: Proposed SLAM framework.

4.1 Basic Visual Localization and Mapping

The environment is represented by a set of landmarks X and camera poses C, located in a global coordinate frame $\mathbf{W}$. The camera pose is $C = [R \mid T]$, where $T_{3\times1}$ is a translation vector and $R_{3\times3}$ is a rotation matrix. Each map point is represented by its homogeneous coordinates $X = [x\; y\; z; 1]^T$. For feature extraction and data association we make use of the well-known SIFT-descriptor proposed by Lowe [21], which is attached to every feature point.
Our SLAM system consists of four main parts highlighted in Figure 2: pose prediction, pose refinement, map update and loop closing. With every new stereo image pair, keypoints are detected in the stereo frames and a point cloud X_c^t in the camera centered coordinate frame is built.
For pose prediction, the valid descriptors are matched against the previous stereo pair to generate a set of corresponding 3D points X_c^{t-1} and X_c^t. The relative motion from C^{t-1} to C^t is estimated by computing a direct least squares solution between the two point sets [12]. To be robust against matching outliers, a RANSAC routine is applied [8]. By knowing C^{t-1} and the relative motion, a predicted pose $\hat{C}^t$ is calculated.
To establish correspondences between the map and the current view $\hat{C}^t$, we perform SIFT-descriptor matching between observed points and map points in the field of view of $\hat{C}^t$.

Again, a RANSAC routine is applied to robustly estimate the $\tilde{C}^t$ relative to the map.
Sparse bundle adjustment (see Appendix 6 of [13]) over the last N frames is used to refine camera pose and map points through minimizing reprojection errors. To update the map, all unmatched features are added.
For loop closure detection, we use the vocabulary tree proposed by Nister *et al.* [23], which has been successfully applied before [9] [3]. Once a loop is detected, sparse bundle adjustment is applied over the entire map.
If global re-localization is required, we perform exhaustive SIFT-matching of the currently observed features against all features of the map. Camera pose is then estimated through a 3-point RANSAC algorithm as used for pose estimation.
A schematic of the mapping procedure is shown in Figure 2. Adaptations of this standard SLAM procedure to incorporate the HOC-descriptor are described in the subsequent section.

4.2 Extension to dynamic map building

A SLAM framework which simply adds every valid feature to the map will fail in a dynamic environment. Either the map becomes too large to handle, or data association will fail because of ambiguities.
We hereby augment the SLAM framework described in the previous section by the HOC-descriptor to address these problems. In this case, the pose correction makes use of visibility information from the HOCs, and the map update includes a HOC update with a map thinning routine (see red highlighted step in Figure 2).
During pose correction, we are able to perform more effective prefiltering by selecting all map points in the view cone with an importance weight exceeding a predefined threshold. Every time a new feature is added, its HOC-descriptor is created and updated according to the refined camera pose. The remaining bins are marked unseen. HOC-descriptors which successfully matched during the pose correction, are upweighted, including those with a low importance weight. Map points which are in the view cone, but did not produce a match, are downweighted.
To keep a constant map size we apply a simple thresholding operation to the reweighted bins. If the maximum of all bins of a descriptor drops below a threshold, the associated landmark is removed from the map. In all of our experiments this threshold was set to $p_{min} = 0.2$.

5 Experiments

We performed a series of synthetic and real-world experiments using a stereo-camera with a baseline of $12cm$ and a resolution of 640×480. We compared the performance of the standard SLAM algorithm with its extension using the HOC-descriptor. In both scenarios we evaluated the map growth over time and the pose estimation error, where groundtruth was available. Regarding parametrization, we always chose 24 histogram bins in our experiments, and adapted parameter λ manually, according to the scene dynamics.

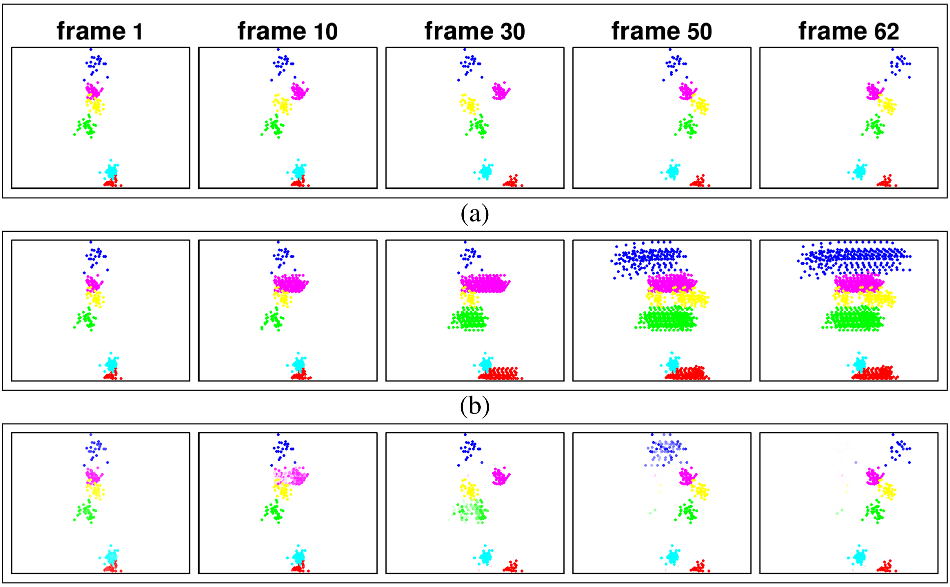
5.1 Synthetic Experiments

We simulated both a static and a moving stereo-camera with a resolution of 640×480 pixels surrounded by static and dynamic objects. Objects are hereby represented as 3D point clouds of variable size covering approximately 10% of the image.

The static camera allows to quickly evaluate localization accuracy and easy visualization.

	translational error [mm]		**rotational error [deg]**		**map size**	
testcase	standard	HOC	standard	HOC	standard	HOC
Simulation 1	12.1/179.1	0.9/41.5	0.46/6.10	0.05/2.41	1526	435
Simulation 2	8.6/79.8	0.2/0.4	0.33/2.80	0.00/0.01	1670	441
Simulation 3	113.5/352.5	0.4/2.1	2.78/8.66	0.00/0.01	3425	825
Simulation 4	12.2/97.2	1.2/3.1	0.15/4.36	0.06/0.13	1024	295

Table 1: Synthetic experiments. Mean/maximal translational and rotational errors were evaluated. The map size after the last frame is also given. Simulations 1, 2, 3 included a static camera with differently moving objects, while Simulation 4 included a moving camera.



(a)

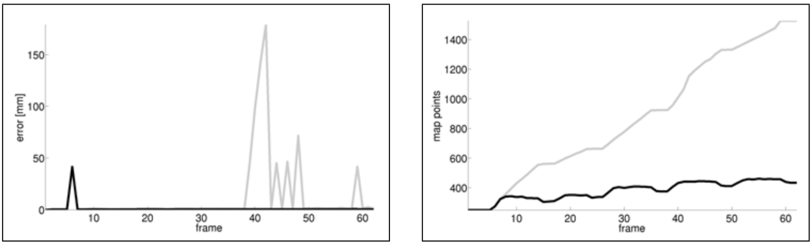
(b)

(c)

Figure 3: Map evolution for Simulation 1. Six objects move in horizontal direction in front of the stereo camera. (a) Some images over time. (b) Backprojected map for standard SLAM procedure over time. (c) Backprojected map for the extended SLAM procedure over time. Saturation encodes the feature weight.

Throughout the experiments with a moving camera, we applied a constant camera velocity of 0.3 m/sec and captured at 25 frames per second. The image measurements, i.e. projections of the 3D points, were corrupted with Gaussian noise ($\sigma = 0.5$). Objects were moving with a constant velocity of 0.5 m/sec lying approximately 80 to 220 cm in front of the camera.

We conducted four synthetic experiments, where three of them assume a static stereo system (Simulation 1 - Simulation 3) and one simulates a translational moving stereo system (Simulation 4). We evaluated the camera pose (translational and rotational error) at every frame, and monitored the map size over time. Some images from Simulation 1 are shown in Figure 3, where the reprojected object points for five frames are presented (see Figure 3(a)), the backprojected map of the standard SLAM algorithm (Figure 3(b)) and the extended al-



(a) (b)

Figure 4: Localization error (a) and map size (b) over time for Simulation 1. Standard SLAM approach (light gray) compared to the extended approach (black).

gorithm (Figure 3(c)). The weight of each landmark is visualized by the saturation of each feature point in Figure 3(c). Using the HOC descriptor, old positions disappear after a few frames. More recent positions of the moving objects are given a higher weight and assist localization. Pose estimation stability and map growing for Simulation 1 over time are presented in Figure 4. The results for all synthetic experiments are summarized in Table 1.

5.2 Real-world Experiments

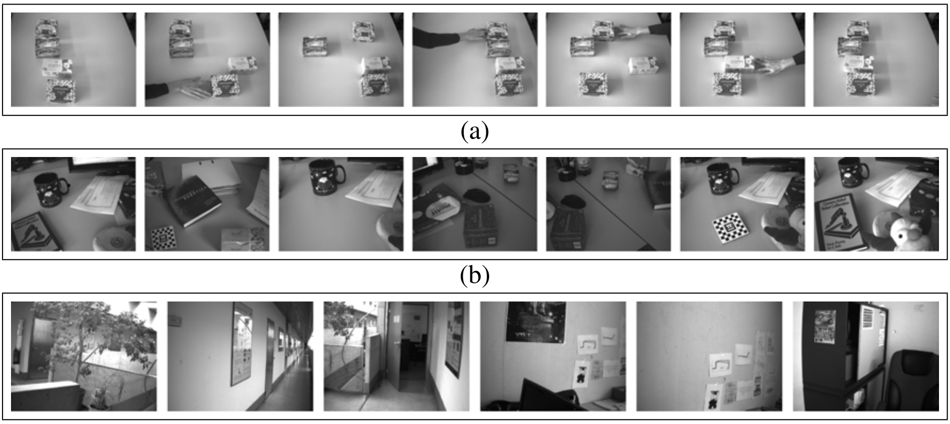
Tests (named Testset 1 - Testset 3) have been made with the stereo-rig mounted statically on a tripod to get groundtruth data for camera pose and relocalization performance in high dynamic scenes. Objects in front of the camera have been moved or deleted (see Figure 5(a)) over a duration of 570, 490 and 276 frames respectively. Table 3 gives the mean and maximum translational and rotational errors for all test sets.

In addition, we evaluated re-localization performance at every 30-th frame with the method described in Section 4. The mean and maximal pose errors are summarized in Table 2 (videos of two localization experiments are shown in the supplementary material named Testset1.avi - Testset2.avi).

Finally, we moved the camera multiple times over an office-table denoted as Testset 4 (see Figure 5(b)) while manipulating dominant objects (remove, occlude or re-appear after some time). A comparison of resulting maps and camera trajectories for the standard and HOC approach can be found in the supplementary material (Testset4Map.jpg). We also acquired larger sequences named Testset 5 and 6 (904 and 1024 frames respectively), covering a $11 \times 7m^2$ flat and a $14 \times 17m^2$ office scene (see Figure 5(c)). In both cases, standard and HOC approach produced comparable camera trajectories although the extended algorithm produced a smaller feature map. Results for the final map size are given in Table 3. The final maps and camera trajectories for Testsets 5 and 6 are shown in the supplementary material (Testset5(6)HoC.avi and Testset5(6)std.avi).

6 Conclusion

The histogram of oriented cameras allows to encode spatial visibility information on a feature basis. In contrast to most existing approaches, which encode visibility in a camera-centered way (e.g. using keyframes), we propose to add a per-feature spatial histogram of the number of observations. Although the amount of saved data per feature is larger, we finally save memory by keeping the overall map size small. In our experiments we reduced memory



(a)

(b)

(c)

Figure 5: Real world experiments. (a) Testset 3 assuming a static camera with various moving objects. (b) Testset 4 taken from a small office scene. (c) Testset 6 covering a larger office environment with opened/closed doors, several occluders and removed objects.

	translational error [mm]		**rotational error [deg]**	
testcase	standard	HOC	standard	HOC
Testset 1	15.9/41.3	6.7/28.5	0.78/1.96	0.31/1.31
Testset 2	187.7/281.2	20.1/76.4	8.86/12,75	1.03/3.53
Testset 3	9.2/17.0	3.5/8.6	0.56/1.10	0.19/0.43

Table 2: Re-localization error for three real-world experiments, assuming a static camera. The mean/maximum rotational and translational errors are presented for both approaches.

consumption from 26% up to 85%. Considering computation time, the histogram update, and the query of a specific histogram bin are comparably cheap.

Tracking accuracy (i.e. determining relative camera motion between subsequent frames) is good with both methods, because ambiguous parts of the map are filtered through the reprojection error. Localization without rough prior knowledge of the pose is more robust with our method, though. Especially after a long time of operation, the standard map becomes filled up with ambiguous data, and correct localization may fail.

An open issue is the undefined behavior after loop closing. To build the histogram we have to assign an orientation to each feature. A bundle adjustment procedure after loop closing might re-orient many feature points and cameras relative to each other, which results in inconsistent histograms. Yet, in our experiments we did not experience a failure of the system due to this effect.

7 Acknowledgement

This work was supported by the Austrian Research Promotion Agency (FFG) and Federal Ministry of Economics, Family Affair and Youth (BMWFJ) within the Austrian research Studio Machine Vision Meets Mobility.

	translational error [mm]		**rotational error [rad]**		**map size**	
testcase	standard	HOC	standard	HOC	standard	HOC
Testset 1	16.5/28.9	5.7/18.1	0.82/1.44	0.28/1.03	7904	816
Testset 2	190.2/281.4	17.6/55.7	8.99/12.94	0.95/3.07	11899	1593
Testset 3	30.7/57.1	2.5/15.7	1.51/2.81	0.11/0.88	6192	855
Testset 4	X	X	X	X	6838	3817
Testset 5	X	X	X	X	8992	1627
Testset 6	X	X	X	X	15120	3680

Table 3: Results for the real world experiments. Mean/maximum pose errors and final map size are given.

References

[1] P. Biber and T. Duckett. Dynamic maps for long-term operation of mobile service robots. In *Robotics: Science and Systems*, pages 17–24, 2005.

[2] M. Chli and A. J. Davison. Automatically and efficiently inferring the hierarchical structure of visual maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 387–394, 2009.

[3] M. Cummins and P. Newman. Accelerated appearance-only SLAM. In *International Conference on Robotics and Automation*, pages 1828 –1833, 2008.

[4] A.J. Davison, I.D. Reid, N.D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *PAMI*, 29(6):1052–1067, 2007.

[5] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In *IEEE International Conference on Robotics and Automation*, pages 1009–1014, 2000.

[6] J.F. Dong, S. Wijesoma, and A.P. Shacklock. Extended rao-blackwellised genetic algorithmic filter SLAM in dynamic environment with raw sensor measurement. In *International Conference on Intelligent Robots and Systems*, pages 1473–1478, 2007.

[7] E. Eade and T. Drummond. Monocular SLAM as a graph of coalesced observations. In *Proc. 11th IEEE Int. Conf. on Computer Vision*, pages 1–8, 2007.

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981.

[9] F. Fraundorfer, C. Wu, J.-M. Frahm, and M. Pollefeys. Visual word based location recognition in 3D models using distance augmented weighting. In *Fourth International Symposium on 3D Data Processing, Visualization and Transmission*, 2008.

[10] A. P. Gee, D. Chekhlov, w. Mayol, and A. Calway. Discovering planes and collapsing the state space in visual slam. In *Proceedings of the 18th British Machine Vision Conference*, 2007.

[11] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. In *IEEE International Conference on Robotics and Automation*, pages 1557–1563, 2003.

[12] R.M. Haralick, H. Joo, C. Lee, X. Zhuang, V.G. Vaidya, and M.B. Kim. Pose estimation from corresponding point data. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6):1426–1446, 1989.

[13] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[14] S. Hochdorfer and C. Schlegel. Towards a robust visual SLAM approach: Addressing the challenge of life-long operation. In *International Conference on Advanced Robotics*, pages 1–6, 2009.

[15] S. Holmes, G. Sibley, G. Klein, and D. W. Murray. A relative frame representation for fixed-time bundle adjustment in SFM. In *IEEE international Conference on Robotics and Automation*, pages 2631–2636, 2009.

[16] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *6th International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007.

[17] G. Klein and D. Murray. Improving the agility of keyframe-based SLAM. In *10th European Conference on Computer Vision*, pages 802–815, 2008.

[18] K. Konolige and M. Agrawal. FrameSLAM: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, 2008.

[19] K. Konolige and J. Bowman. Towards lifelong visual maps. In *International Conference on Intelligent Robots and Systems*, pages 1156–1163, 2009.

[20] G. Lidoris, D. Wollherr, and M. Buss. Bayesian state estimation and behavior selection for autonomous robotic exploration in dynamic environments. In *International Conference on Intelligent Robots and Systems*, pages 1299–1306, 2008.

[21] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[22] D. Migliore, R. Rigamonti, D. Marzorati, M. Matteucci, and D. G. Sorrenti. Use a single camera for simultaneous localization and mapping with mobile object tracking in dynamic environments. In *ICRA Workshop on Safe navigation in open and dynamic environments Application to autonomous vehicles*, pages 27–32, 2009.

[23] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.

[24] L.M. Paz, P. Pinies, J.D. Tardos, and J. Neira. Large-scale 6-DOF SLAM with stereo-in-hand. *IEEE Transactions on Robotics*, 24(5):946–957, 2008.

[25] D. Schleicher, L.M. Bergasa, R. Barea, E. Lopez, M. Ocaa, J. Nuevo, and P. Fernandez. Real-time stereo visual SLAM in large-scale environments based on SIFT fingerprints. In *IEEE International Symposium on Intelligent Signal Processing*, pages 1–6, 2007.

[26] S. Shojaeipour, S. M. Haris, K. Khalili, and A. Shojaeipour. Motion planning for mobile robot navigation using combine quad-tree decomposition and voronoi diagrams. In *International Conference on Computer and Automation Engineering*, pages 90–93, 2010.

[27] G. Sibley, C. Mei, I. Reid, and P. Newman. Adaptive relative bundle adjustment. In *Robotics Science and Systems*, 2009.

[28] J. Sola. *Towards Visual Localization, Mapping and Moving Objects Tracking by a Mobile Robot: a Geometric and Probabilistic Approach*. PhD thesis, Institut National Politechnique de Toulouse, 2007.

[29] T.-D. Vu, J. Burlet, and O. Aycard. Grid-based localization and online mapping with moving objects detection and tracking: new results. In *IEEE Intelligent Vehicles Symposium*, pages 684–689, 2008.

[30] C.-C. Wang and C. Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *IEEE International Conference on Robotics and Automation*, pages 2918–2924, 2002.

[31] J. Yang, Z. Qu, J. Wang, and K. Conrad. Comparison of optimal solutions to real-time path planning for a mobile vehicle. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, pages 1–11, 2010.