ROBOTICS NAVIGATION USING MPEG CDVS

Mohit[1], Mukul[2], Indu Khatri[3]
[1,2]Student, [3]Guide Computer Science,BMCEM,Sonipat,India

***Abstract:* The choice for image descriptor in a visual navigation system is not straightforward. Descriptors must be distinctive enough to allow for correct localization while still offering low matching complexity and short descriptor size for real-time applications. MPEG Compact Descriptor for Visual Search is a low complexity image descriptor that offers several levels of compromises between descriptor distinctiveness and size. In this work we describe how these trade-offs can be used for efficient loop-detection in a typical indoor environment.**

I. INTRODUCTION

Robots that navigate through unknown environments, such as autonomous vacuum cleaners, all face a common challenge: to create a representation of their environment while simultaneously trying to locate themselves. This problem is known in literature as Simultaneous Localization and Mapping (SLAM) and its formulation has been thoroughly reviewed in [1] and [2]. Approaches to SLAM usually involve two alternate phases. During Motion Prediction the robot uses internal parameters to estimate local displacements, while during Measurement Update the robot interacts with the environment to improve both its pose estimation as well as its estimate of its environments map.

In visual-based SLAM, motion prediction can be obtained by extracting and matching visual features from a sequence of images in a process called feature-based visual odometry[3]. These same features can also be used as landmarks of for loop- detection during the measurement update phase in a complete Visual SLAM (VSLAM) system [4].

The purpose of Loop-detection is to identify landmarks on the map that have already been seen by the robot during early stages of navigation. During this process, if a particular landmark is allegedly found in two different places it means that the estimated trajectory described by the robot has probably drifted at some point. The relative displacement between these two appearances can then be used by a global optimizer to improve estimations of both the landmarks' positions as well as robot's pose.

Early approaches to loop-detection using visual features include the work in [5], where authors used the SIFT [6] for its distinctive power and thus capability of correctly find a loop. SIFT's distinctiveness; however, comes at a high price in terms of compute complexity leading to substantial battery consumption. Moreover, the amount of SIFT features generated by a single image also makes it prohibitively expensive in terms of bandwidth requirement where remote processing is needed such as in collaborative mapping. In [7] authors have used an intermediate level of representation to speed-up loop detection known as *bags of visual words* [8]; a technique originally developed to compare similarity between documents and which is still considered the state of the art today. Finally as the robot navigates throughout its environment the number of observed landmarks increases and so does the number of descriptors it stores for loop-detection. This means that loop-detection algorithms are bound to become expensive in terms of both memory and computational complexity [2] as the map grows. This forces system designers to either choose less complex descriptors, risking wrong data association, or to overestimate memory demands during hardware design. The problem of finding a perfect balance between de- scriptor distinctiveness and descriptor size is not exclusive to the VSLAM domain. When dealing with large databases, Content-Based Image Retrieval (CBIR) systems face this very same issue. Very recently, the Moving Picture Experts Group (MPEG) has defined a new industry standard for CBIR known as Compact Descriptors for Visual Search (MPEG CDVS). The standard specifies various modes of compression that offer trade-offs between distinctiveness and size and also provides with suggested metrics for image comparison that quantify how similar two images are. In this work we claim that the characteristics that make MPEG CDVS a good descriptor for CBIR, also make it ideal for robotic navigation. More specifically, we state that MPEG CDVS can be used as a fast, reliable and storage-efficient loop detector in a typical indoor VSLAM application. Our first contribution comes in Section III where we de- scribe a probabilistic approach to loop detection using the standard's suggested similarity metric. We then compare per- formance of CDVS compression modes in terms of matching speed, feature extraction and storage requirements with the well-known SIFT descriptor for five different types of indoor floors and show that CDVS has superior performance in all cases in Section IV. Finally, in Section V we apply our proposed method to a real robotic application and show that our VSLAM approach gives better results than state-of-the-art laser-based SLAM.

II. THE MPEG COMPACT DESCRIPTOR FOR VISUAL SEARCH

The Compact Descriptor for Visual Search (CDVS) is the new standard for Content Based Image Retrieval developed by the Moving Picture Experts Group [10]. It defines an image description tool designed for efficient and interoperable visual search applications.

A. Descriptor Generation

A CDVS descriptor is made of two parts: one global descriptor associated to the entire image and a set of local descriptors associated to specific points in the image known as interest points. The entire descriptor extraction process can be summarized as follows:

1) Interest Point Detection;

2) Feature Selection and Descriptor Extraction; where based on the level of compression used only a limited number of interest points will account for the final descriptor.
3) Local Descriptor Extraction;
4) Local Descriptor Compression;
5) Coordinate Coding;

Global Descriptor Generation: It is an aggregation of lo- cal descriptors to generate a fixed, small size description of the entire image. The final result of CDVS extraction is a compressed file whose size is upper-bounded by 512B, 1kB, 2kB, 4kB, 8kB, and 16kB associated to the extraction modes 1 to 6 respec- tively.

B. Descriptor Matching and Score

When comparing two images MPEG CDVS suggests the use of two different types of matching scores: global score and a locals score. The global score is given as a weighted correlation between the two images global descriptors. The local score given between two images results from the sum of local scores of each descriptor in those images, i.e a one-to-one comparison is made between local descriptors from the two images. Finally, the standard also suggest the use of a geometric consistency analysis, known as Distrat [11], to eliminate false matches between descriptors based on their geometry disposition. In order to detect a loop as defined in III-A, we consider only those features that have passed also the geometric con- sistency test. Moreover we consider the values given by local score as our means to indirectly measure the probability of loop detection for it gives more reliable results..

II. PROPOSED MOTION MODEL AND LOOP DETECTION

A robot carrying a calibrated camera navigates through an indoor environment while taking a picture I_k of the floor below at each time step k. The robot's starting position and heading define both origin and x-axis of a global coordinate frame. This coordinate system then becomes uniquely defined as we choose the z-axis to point upwards.

We assume the environment's floor to be a planar surface so that, for each time step $k > 0$, the robot's pose is given by $\mathbf{x}_k = [x_k, y_k, \theta_k]^T$, where x_k and y_k indicate the robot's coordinates and θ_k is the robot's heading.

Final motion between time steps k 1 and k can be modeled as a rotation followed by translation, so that at t = k pose can be recursively obtained as

$$[x_k, y_k]^T = R_{(\Delta\theta_{k-1,k})}[x_{k-1}, y_{k-1}]^T + T_{k-1,k} \quad (1)$$

$$\theta_k = \theta_{k-1} + \Delta\theta_{k-1,k} \quad (2)$$

where $\Delta\theta_{k-1,k}$ is the rotation angle estimated between time steps $k-1$ and k, $R_{(\Delta\theta_{k-1,k})}$ is the rotation matrix for that same angle, and $T_{k-1,k}$ is the translation vector.

A. Loop Definition

The use of a downward-facing camera allows for a natural definition of loop based on the intersection of imaged regions. For images I_a and I_b taken along the robot's path, we define loop as a function of the overlap ratio between the floor area observed by these two images. So given the area of intersection area(I_a I_b), and the respective area of union area(I_a I_b), a loop can be defined as

$$J = \frac{\text{area}(I_a \cap I_b)}{\text{area}(I_a \cup I_b)} \quad (3)$$

$$\text{loop}(I_a, I_b, r) = \begin{cases} 1 & \text{if } J \geq r \\ 0 & \text{if } J < r \end{cases} \quad (4)$$

where r is the threshold that defines the minimum overlap ratio for which two intersecting images can be considered a loop. In this work we set this threshold to r = 0.33, which roughly amounts for an area intersection of 50% when I_a and I_b have the same areas.

B. Loop Probability

Loop detection as defined in (4) requires the knowledge of how much intersection there is between the two images. In order to indirectly measure the probability of having a particular area ratio we use the *local score* given between two images so that

$$P(\text{loop} = 1|\text{score} = s) = P(J \geq r|\text{score} = s) \quad (5)$$

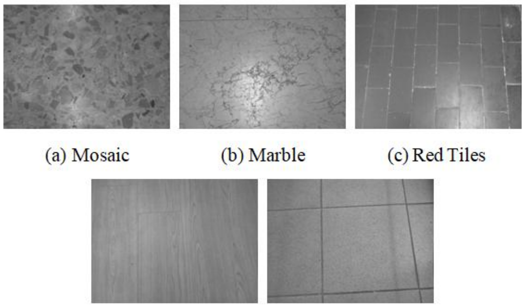
$$P(J \geq |r\ \text{score} = s) = \frac{P(J \geq r, \text{score} = s)}{P(\text{score} = s)} \quad (6)$$

The conditional probability in (5) can be experimentally estimated through (6) by combining the knowledge of the cam- era's parameters with a source of relative motion estimation. This process will be described in depth during the next section.

III. TRAINING OF PROPOSED MODEL

Besides being distinctive, a descriptor needs also to be economical in terms of storage and extraction and matching times in order for it to be considered as a feasible option for loop detection.

In this section we analyze the distinctiveness of all CDVS' compression modes for the five different types of floorings seen in figure 1. We also compare their memory and processing time requirements with a popular implementation of the SIFT descriptor found in [12].



(a) Mosaic (b) Marble (c) Red Tiles

(d) Printed Wood (e) Dotted Tiles

Fig. 1: Different types of floorings commonly found in indoor environments.

A. Distinctiveness of CDVS local score

Our analysis starts by driving the robot forward for 10 meter using a PointGrey Grasshopper 3 camera rigidly mounted on a Turtlebot 2 in a setup defined in section III.

For each floor type we extract CDVS descriptors the sequence of images and match each image with all the previous ones using CDVS *local score* to measure similarity. We repeat this process for all modes of compression to evaluate its effect on feature distinctiveness.

Distinctiveness in this context means to have high *local score* for images having overlapping regions and very low scores otherwise. Since images were taken in sequence during robotic motion, images that are close in the sequence are also spatially next to each other, and thus should have high local score.

A visual representations of these matches using compression mode 6 is given in figure 2 where pixel intensities in position (i, j) represent the local score between current image i and a previously visited image j. Since we only match current images with previous ones, each matrix representing the matches is triangular.

To allow for a fair visual comparison, the matrices values have been normalized. Yellow pixels mean high local score while dark blue pixels indicate a low score. The presence of small, bright triangles seen at the lower end of each matrix indicates when the robot stopped.

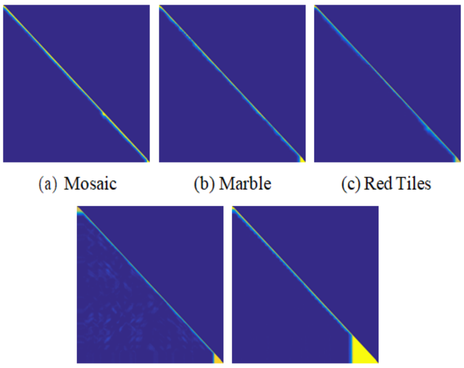

(a) Mosaic (b) Marble (c) Red Tiles
(d) Printed Wood (e) Dotted Tiles

Fig. 2: Visual representation of Local Score for different floor types.

Ideally, these matching matrices should display increasingly intensity of pixel values (yellow) in regions near each diagonal and very low values (dark blue) everywhere else. The natural randomness intrinsically associated to the production of most of the floor types enables them to have a relatively the thick principal diagonals and to display very low matching scores where no overlap occurs.

The one noticeable exception occurs for the printed wood floor. This particular artificial type flooring is made of a printed repetitive patterns. The effect of such patterns appears as bright spots on its matching matrix and can be particularly harmful for loop-detection since it leads to erroneously detected loops.

We can observe the evolution of these spots and the diagonal thickness in figure 3 as we vary the compression mode.

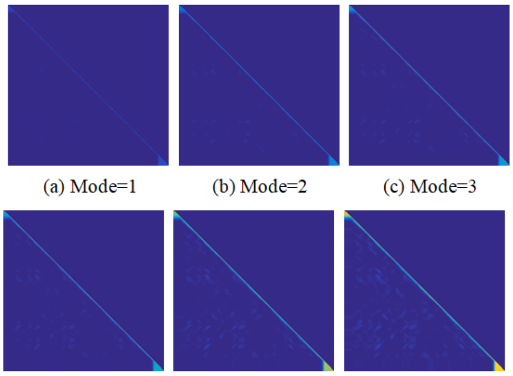

(a) Mode=1 (b) Mode=2 (c) Mode=3
(d) Mode=4 (e) Mode=5 (f) Mode=6

Fig. 3: Visual representation of Local Score for the Printed Wood floor using different compression modes.

It is clear that the diagonal thickness decreases as the compression level increases, i.e. for lower modes of compres- sion. This phenomenon happens to all flooring types and it is due to the fact that CDVS will use fewer keypoints with shorter local descriptors to represent each image. This makes it difficult to correctly match images that are even just slightly displaced with respect to one another. Therefore; as expected, lower modes of compression can be considered to offer less distinctive local descriptors.

On the other hand and for the same reason, bright spots on the wooden pattern become even more visible as the level of compression increases, which makes this particular kind of flooring the worst case scenario and also our study case to test CDVS for loop detection.

B. Effects of Feature Selection

Besides being able to correctly distinguish between different floor patches, CDVS must also be economical in terms of storage, extraction time and matching time if it is to be considered as an alternative to well-established descriptors such as SIFT [6]. Here we examine these characteristics by analyzing the same five types of flooring.

As seen in figure 4, feature selection has the effect of reducing the number of local features generated for each image. Since the final binary size of a CDVS descriptor is limited by its compression mode, the maximum number of local descriptors produced by each mode is upper-bounded and does not significantly depend on the particular type of flooring.

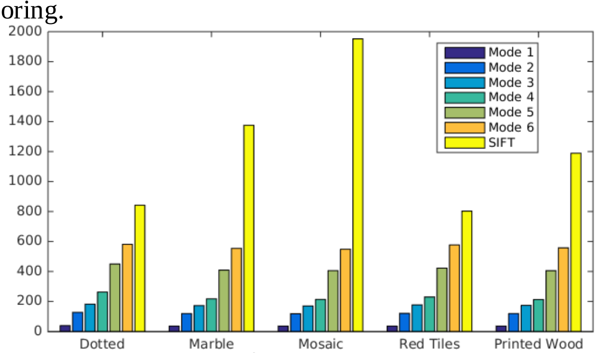


Fig. 4: Average number of extracted local descriptors per image for each type of flooring.

In terms of memory efficiency, feature selection has a clear effect on reducing storage requirements. For example, an image taken from a Mosaic floor would normally require over 300kB of memory if SIFT descriptor were to be used, considering implementations such as [12], while CDVS would require at most 16kB at its least compressed mode.

Another positive effect of feature selection is the reduction of extraction time as reported in table I. Since feature selection is made based on keypoints' characteristics, only features from selected keypoints will be processed. Moreover, having a limited number of descriptors per image will also limit the time spent for comparing two images as reported in table II. Finally we observe that both extraction and matching times are at least an order of magnitude lower than SIFT and that these values show little variation within a given compression mode.

Having upper-bounded memory requirements and extraction and matching times that are relatively invariant to the different types of floorings are essential qualities for systems that may work on different environments. For example, system requirements for automatic vacuum cleaner should not depend on consumer's specific type of floor.

floor types	*mode 1*	*mode 2*	*mode 3*	*mode 4*	*mode 5*	*mode 6*	*SIFT*
Dotted Tiles	16.2	15.4	15.5	16.2	18.9	21.0	217
Marble	15.6	15.3	15.3	16.3	18.9	21.4	295
Mosaic	15.9	15.8	16.0	18.9	22.4	22.3	388
Red Tiles	14.6	14.8	14.7	15.5	18.1	21.0	209
Printed Wood	15.2	15.2	15.3	16.0	18.8	21.0	270

TABLE I: Average extraction times per image in milliseconds for each CDVS mode of compression and SIFT.

C. Estimating Loop Probability

A camera's intrinsic and extrinsic parameters define the camera's pose with respect to the world and also allow us to make real world measurements directly from images. These Relative motion during training was obtained using the robot's odometry, and although odometry suffers from error accumulation after long trajectories, it does provide dependable relative motion estimations over short range distances. Moreover, images that are relatively distant from each other, will have zero overlapping region an therefore error accumulation will constitute a problem. During training phase relative motion was obtained by using a Kalman filter that combined information from both wheel odometry and a robot's internal gyroscope during the experiment described at the beginning of this section.

By combining these pieces of information with the local scores of each analyzed matching pair, we can generate for each compression mode a loop detection probability curve as defined in 6. The resulting curves as seen in 5 show the probability two images having more than 50% of intersection for each mode given a local score.

Lower compression modes achieve certainty at lower values of local score. This is due to the fact that low compression modes also have fewer descriptors to be used during match.

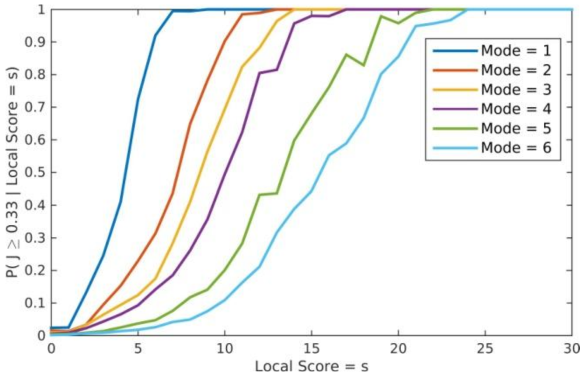


Fig. 5: Conditional loop probability for printed wood floor.

From these curves we select the minimum values values of local score s that guarantee loop detection for each compression mode. These hypothesis values are reported in table III and used to define the loops during the final experiments discussed in section V.

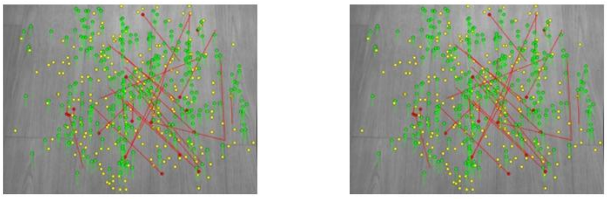
D. Visual Odometry for Testing

In order to demonstrate that our approach could be applied to a vision-only navigation system having no other sensors such as gyroscope or wheel encoder, we have decided to implement VSLAM also using visual odometry. Our robot setup follows the one in [9]. However, although we do use a similar approach to obtain odometry, our main concern in this work is the correct detection of loops for VSLAM.

Depending on system requirements, less complex feature descriptors such as [13] and [14] could be used to generate odometry, while CDVS would be used just for loop detection. However, since local features from each image will already be available, we choose to use CDVS local descriptor to generate visual odometry as well.

For each pair of consecutive images I_{k-1} and I_k we perform a feature extraction and match of MPEG CDVS descriptors, which results into two sets of $N > 2$ matching coordinate pairs. We combine these pixel coordinates with the camera's calibration information and produce the sets P_{k-1} and P_k each containing the 3D coordinates for the N matching pairs. By defining P_{k-1} and P_k to be the centroids of P_{k-1} and P_k respectively, we retrieve rotation and translation using Singular Vector Decomposition.

A visual representation of this process is shown in figure 6.



(a) t=k

(b) t = k + 1

Fig. 6: Matching between images at time k and $k + 1$. Keypoints are indicated as dots. Yellow dots represent non-matching features. Green lines represent correspondences between matches. Red lines are false matches found by Distrat.

Although CDVS already performs geometric consistency validation, we make useof a few RANSAC [15] cycles to remove possible possible remaining outliers and improve results.

VI. EXPERIMENTAL RESULTS

Partial results from Sec. IV have lead us to try our loop-detection technique on the most challenging flooring for loop- closure, i.e. the flooring most susceptible false-loop detection. In this experiment, the robot navigates through indoor office for about 110 meter while taking a total of 7154 images of its printed wood floor and performing loops before finally going
back to its original position.

We first use the sequence of images to generate the path’s visual odometry as described in IV for all except the first compression mode, which was unable to generate enough matching points between consecutive images. For those modes capable of estimating translation and rotation from consecutive images, we report their respective paths in Fig. 7 where we use the room’s blueprint as reference map.

local score	*mode 1*	*mode 2*	*mode 3*	*mode 4*	*mode 5*	*mode 6*
Hypothesis	10	14	15	18	23	25
Experimental	–	20	16	18	24	27

TABLE III: Hypothesized and Experiemtal threshold values for local score loop detection.

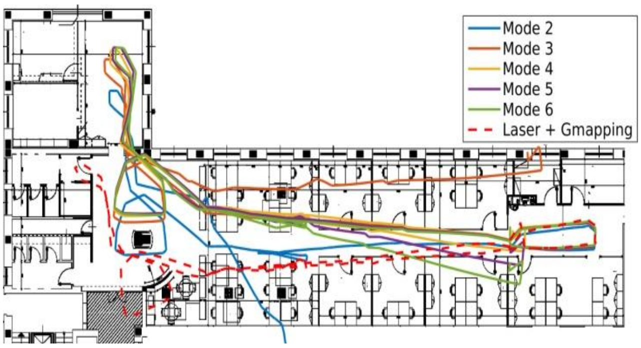


Fig. 7: Path comparison using visual odometry.

We then perform loop detection as described in Sec IV where for each image pair whose local score was above the hypothesized in table III a loop was declared.

For each compression mode, we have represented data from visual odometry and loop constraints as a path graph so that the robot’s trajectory could be optimized using the LAGO graph optimization software [16], whose purpose is to find a coherent sequence of poses that better describe all loop and odomtery constraints, and thus perform VSLAM.

During these experiments, we have observed that the pro-posed local scores thresholds loop-detection found earlier were slightly too permissive and still allowed for small amount of false-positive loops to be detected. This fact has led us to empirically increase these thresholds until reasonable results were obtained. We report these new values as the *Experimental* entries in III, which differ very little from the hypothesized ones and thus proving that the method is still valid. Th resulting trajectories for each compression mode using the experimental thresholds can be seen in Fig. 8.

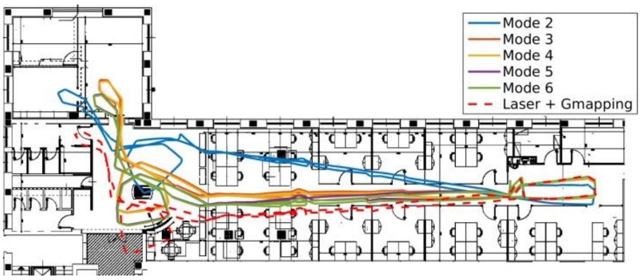


Fig. 8: Paths optimized using LAGO.

A visual inspection between the two figures reveals the improvements obtained for all compression modes when loops are correctly detected. Except for compression mode 2, all improved trajectories pass through the hallway, enter and exit the northwest room and respect the physical constraints present in the map. However, in order to have a more quantitative measure of such improvements we report in III the pose difference between starting and ending poses in the trajectory, which ideally should be none.

	Visual Odometry			*Visual SLAM*		
	Δx (m)	Δy (m)	Δθ (rad)	Δx (m)	Δy (m)	Δθ (rad)
Mode 2	17.35	-6.58	-0.86	0.0725	-0.0088	0.0075
Mode 3	-4.36	1.27	0.03	0.0355	-0.0115	0.0001
Mode 4	0.22	0.19	-0.13	0.0359	-0.0149	0.0086
Mode 5	1.01	0.09	-0.17	0.0302	-0.0011	-0.0249
Mode 6	2.10	0.00	-0.23	0.0221	-0.0056	-0.0128

TABLE IV: Relative pose errors between staring and final position for both visual odometry and VSLAM

property	*mode 2*	*mode 3*	*mode 4*	*mode 5*	*mode 6*	*SIFT*
Storage (MB)	7.67	14.63	28.59	56.55	112.43	1213.84
Time (s)	4.23	6.62	9.62	27.27	58.32	1264.20

TABLE V: Storage requirement for all 7154 images and total matching time between last sequence image and all previous ones

To highlight the gains in terms of both storage savings and matching times with respect to SIFT, we have compared the amount of memory required to save descriptors for all 7154 images using each compression mode and also report the time necessary to compare the last image in the sequence with all previous one. We report these values in V.

Finally, in order to compare our proposed method with existing state of the art frameworks for indoor SLAM, we also report on both figures the path generated using a Hoyuko laser- scanner optimized with the widely used Gmapping algorithm [17].

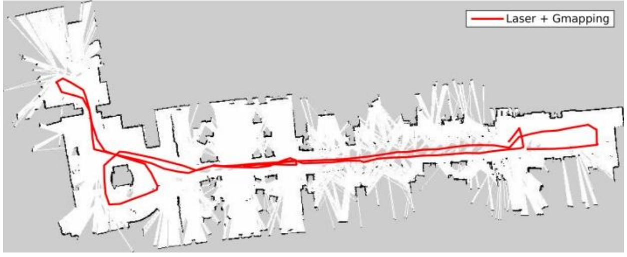


Fig. 9: Map and path generated using a laser scanner with Gmapping algorithm.

At first sight, results from laser scanner can be considered incorrect and unreliable. This occurs because laser scanner was unable to create a precise map of environment and thus was unable to reproduce its path correctly on the real world map. This becomes evident in figure 9 where the path generated by the laser seems to be coherent to its self-generated "bended" map. Our method clearly does not suffer from the same issue.

V. CONCLUSION

In this work we have proposed the use of MPEG CDVS in a SLAM framework for loop-detection in an indoor environment. We have shown experimentally that CDVS' feature selection serves not only to reduce the final descriptor size but also to significantly speed up feature extraction and matching. In our practical experiment CDVS's least compressed mode was shown to be over 20 times faster than SIFT during matching time and to require 10 times less storage space and still able to provide for correct loop-detection. Finally, when we compared to a laser scanner, we have seen that our approach has generated far better results.

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and map- ping: part i," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, 2006.

[2] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and map- ping (slam): Part ii," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[3] D. Scaramuzza and F. Fraundorfer, "Visual Odometry [Tutorial]," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, Dec. 2011.

[4] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii: Matching, robustness, optimization, and applications," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 2, pp. 78–90, 2012.

[5] P. Newman and K. Ho, "Slam-loop closing with visually salient fea- tures," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 635–642.

[6] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE interna- tional conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[7] J. Wang, H. Zha, and R. Cipolla, "Coarse-to-fine vision-based lo- calization by indexing scale-invariant features," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 2, pp. 413–422, 2006.

[8] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, no. 1-22. Prague, 2004, pp. 1–2.

[9] ISO/IEC JTC 1/SC 29/WG 11 (MPEG), *Information technology – Multimedia content description interface – Part 13: Compact descriptors for visual search*, ISO/IEC Std.

[10] S. Lepsoy, G. Francini, G. Cordara, and P. P. de Gusmao, "Statistical modelling of outliers for fast visual search," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–6.

[11] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," 2008, http://www.vlfeat.org/.

[12] H. W. H. Wang, K. Y. K. Yuan, W. Z. W. Zou, and Q. Z. Q. Zhou, "Visual odometry based on locally planar ground assumption," *2005 IEEE International Conference on Information Acquisition*, pp. 59–64, 2005.

[13] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision." in *IJCAI*, vol. 81, 1981, pp. 674– 679.

[14] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2548–2555.

[15] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[16] R. R. G. P. di Torino, "LAGO: Linear approximation for graph opti- mization," https://github.com/rrg-polito/lago, 2000–2004.

[17] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *Robotics, IEEE Transactions on*, vol. 23, no. 1, pp. 34–46, 2007.