2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

A Combined RGB and Depth Descriptor for SLAM with Humanoids

Rasha Sheikh Stefan Oßwald Maren Bennewitz

***Abstract*— In this paper, we present a visual simultaneous localization and mapping (SLAM) system for humanoid robots. We introduce a new binary descriptor called DLab that exploits the combined information of color, depth, and intensity to achieve robustness with respect to uniqueness, reproducibility, and stability. We use DLab within ORB-SLAM, where we replaced the place recognition module with a modification of FAB-MAP that works with newly built codebooks using our binary descriptor. In experiments carried out in simulation and with a real Nao humanoid equipped with an RGB-D camera, we show that DLab has a superior performance in comparison to other descriptors. The application to feature tracking and place recognition reveal that the new descriptor is able to reliably track features even in sequences with seriously blurred images and that it has a higher percentage of correctly identified similar images. As a result, our new visual SLAM system has a lower absolute trajectory error in comparison to ORB-SLAM and is able to accurately track the robot's trajectory.**

I. INTRODUCTION

In the past years, several feature descriptors have been developed. Some of them use textural information such as SIFT, SURF, and more recently ORB [1]. Others are geometric descriptors built from point clouds. A comparison of different 3D descriptors is provided in the work of Hänsch *et al.* [2]. Combining information from different sources such as RGB and depth values increases the robustness of feature descriptors as different cues complement each other, e.g., CSHOT [3], BRAND [4], BAG [5], and SBP [6] use texture and depth information in their descriptors. While BRAND has been shown to have superior performance, it suffers from ambiguity in how texture and shape information are combined.

In this paper, we present a robust visual SLAM system for humanoid robots such as the Nao robot (see Fig. 1) that uses the *combined color and depth* data acquired with an RGB-D camera. We apply an extension of the ORB-SLAM system [7], which is a state-of-the-art system, with the following modifications. (1) We developed a new binary descriptor, DLab, that uses color, intensity, and depth information and use it in place of ORB. To improve the tracking behavior of ORB-SLAM, we (2) use brute force matching between consecutive images and filter out outliers with RANSAC and (3) keep a sequence of previous images together with their estimated camera poses to determine the transformation of the current camera frame. (4) We replaced the place recognizer DBoW2 [8] with a modification of FAB-MAP [9] so that it works with our new binary descriptor, which results in a faster performance in comparison to using SURF as in the original implementation and yields a higher number of correct loop closures compared to using DBoW2.

All authors are with the Humanoid Robots Lab, University of Bonn, Germany.

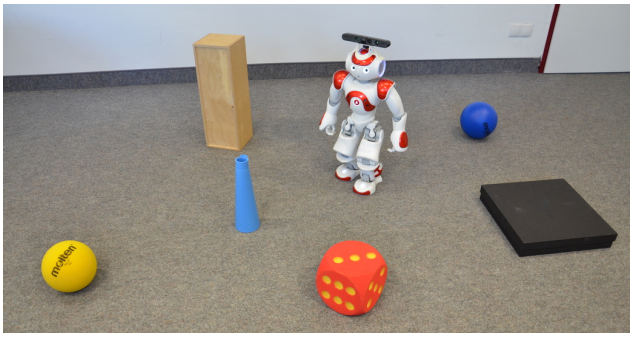

Fig. 1. Walking Nao with an RGB-D camera mounted on its head. Our new descriptor, DLab, combines depth and color information and is used for SLAM.

Our experiments demonstrate that DLab has comparable or better precision at the highest recall under translation and orientation image transformations. In the experimental evaluation, we show that our SLAM system is able to reliably track features even in sequences containing blurry images, which frequently occur when acquiring images with a walking humanoid. Additionally, our new place recognition module has a higher percentage of correctly identified matches. Accordingly, the resulting absolute trajectory error is seriously reduced in comparison to ORB-SLAM.

In summary, our main contributions are the following. First, we propose a new binary image descriptor that makes use of color and depth information and employ it in a state-of-the-art visual SLAM system to improve feature matching. Second, we modified the place recognition module to work with our binary descriptor using newly built codebooks. Lastly, we show in various comparative experiments that we can improve the tracking behavior during visual SLAM.

II. RELATED WORK

A. RGB-D SLAM

Henry *et al.* [10] and Endres *et al.* [11] were among the first who developed a 3D mapping system for data acquired with an RGB-D camera. The general idea of these approaches is to combine the matching of features with pose optimization to reduce the error in the estimates after loop closures. Mur-Artal *et al.* [7], [12] proposed ORB-SLAM, which performs mapping, tracking, relocalization, and loop closure in real-time using ORB features [1]. Figueroa *et al.* [13] combined visual odometry and KinectFusion [14] to reconstruct indoor scenes using the BRAND descriptor [4].

In our work, we apply a modified version of ORB-SLAM, which tracks sparse features and therefore is not as

computationally expensive as methods that run on GPUs [15], [13], [16]. We replaced the ORB descriptor with DLab and enhanced the tracking behavior so it can also handle images with poor features, and can recover from cases where the current camera transformation cannot be determined.

DLab is based on BRAND [4] but we separate the appearance and depth information as mixing them causes ambiguity. Additionally, we change the way depth information is used. We do not use the point clouds and perform fast pixel tests on patches to speed-up the construction of the descriptor.

B. Appearance-Based Loop Closing

Cummins *et al.* developed FAB-MAP [17], which is an appearance-based approach for mapping. It uses the bag-of-words model to decide whether a place is a new location or has been visited before. Hereby, the system uses the observation that some features are more likely to appear together rather than separately.

Gálvez-López and Tardos presented DBoW2 [8], a fast place recognizer that is also based on the bag-of-words approach. To speed up the feature extraction step, the authors initially used the binary feature descriptor BRIEF. Mur-Artal and Tardós then modified DBoW2 to use ORB features [18], which are rotation and scale invariant. ORB-SLAM [7] uses DBoW2 as its place recognition module. In the experiments with our Nao robot, we experienced that a lot of the images it identifies as similar are in fact of different places. We therefore replaced DBoW2 with FAB-MAP [9] and modified it so that it works with DLab. Since the new descriptor is binary, it is also faster compared to when using SURF, the descriptor originally used by FAB-MAP.

Sünderhauf *et al.* have presented a method that relies on CNNs for feature extraction, but it is computationally more expensive and requires processing on GPU [19].

C. SLAM and Visual Odometry for Humanoid Robots

Stasse *et al.* [20] presented a 3D SLAM system for humanoid robots. The authors combined data from the robot's walking pattern generator with odometry, IMU data, and visual features from a monocular camera in an EKF framework. Since images taken by humanoid robots can suffer from blur due to the swaying motion during walking, Pretto *et al.* [21] investigated how to mitigate that effect by developing an approach that only selects highly distinctive features.

Oriolo *et al.* [22] used the head pose provided by the PTAM algorithm and the torso orientation from the IMU measurements in the correction step of EKF to localize a humanoid robot.

In our work, we use the RGB and depth information to simultaneously localize the robot and map the environment. We match features of consecutive images and apply bundle adjustment to optimize the pose of the robot. We use distinctive frames, i.e., images with their estimated camera poses, to build a pool of locations for the place recognition module, which indicates whether a loop closure is likely to have appeared.

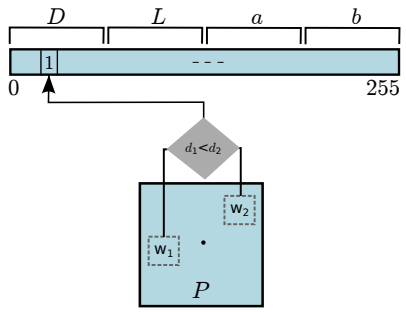

Fig. 2. Layout of our descriptor. Each field in the binary descriptor is filled out by comparing one pair of windows in a patch P. The descriptor is 256 bits long and has equal sections for the depth, intensity, and color channels.

III. DLab Feature Descriptor

Having a good descriptor that is capable of capturing the information in a keypoint patch is essential for good feature matching in visual SLAM. The speed of creating a new descriptor is also of importance since in order for the SLAM system to perform well, a lot of image features need to be computed at every time step. We therefore opted to use binary descriptors which are fast as they carry out simple pixel tests and matching them can also be performed quickly by using the Hamming distance for comparison which is basically an XOR operation.

DLab builds upon BRAND [4], which combines the appearance and shape information by performing the logical operation OR on the result of appearance and shape tests. This produces ambiguity when comparing patches, since the system for example can erroneously conclude that two regions are similar although they only agree on the appearance but not on the depth.

Therefore, we build the 256 bit descriptor out of four equally sized parts: One for the depth tests and one for the intensity, respectively, and in addition include color information. Although intensity is often used to mitigate the effect of noise and illumination changes, it inevitably does not carry as much information as the original color channels. Thus, we combine the advantages of both and use the last two parts of the descriptor for color information.

Our system first converts the image from the RGB to the Lab color space providing an intensity and two color channels and then computes integral images for those three channels and for the depth channel. The size of the patch and the scale for each keypoint is then computed as in the original implementation. After rotating and scaling the patch, pixel tests are performed to fill the 256 bit binary string. To minimize the impact of noise, the tests are carried out on pairs of windows that are 9 pixels wide and we use the precomputed integral images. The location of the windows are precomputed using a Gaussian distribution as in BRAND [4].

Fig. 2 shows how the descriptor for a patch of pixels P centered around a keypoint is constructed. The integral sum value of the depth values in window w_1 denoted as d_1 is compared to that of the second window w_2 denoted as d_2. If the test is positive, the associated bit in the binary vector is

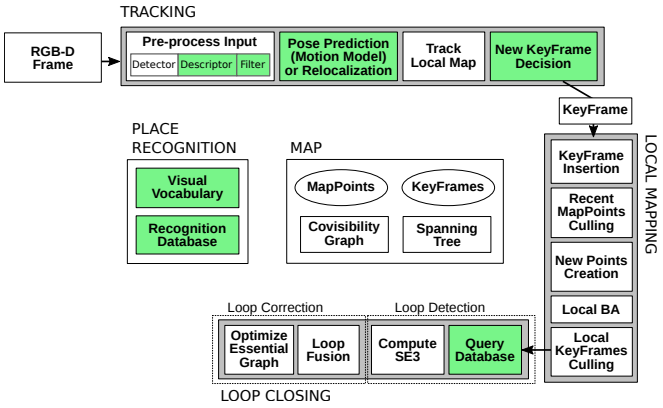

Fig. 3. Overview of the ORB-SLAM system. The figure is from [12] with the changed components marked in green.

set to 1. The same steps are followed to fill out the intensity L and the colors a and b test parts of the descriptor. The source code can be found at `https://www.hrl.uni-bonn.de/research/DLab`.

IV. VISUAL SLAM

Fig. 3 shows an overview of the ORB-SLAM system with the components we changed highlighted in green.

A. Pose Tracking

ORB-SLAM tracks the position of the camera frame by using the motion computed from the poses of the previous two frames. After the frame is initialized to a new pose, the map points seen in the previous frame are projected into the current frame and the search for matches is carried out by comparing descriptors in a window around the projected point. These matches are then used for pose optimization with $\mathbf{g^2o}$ [23] using a variant of local bundle adjustment where the points are fixed and the pose is optimized.

When there are rotations in the movement of the robot or in case of a jittering head with the RGB-D camera during walking, the motion model frequently fails at providing a good initial estimate. Accordingly, no good matches can be found and tracking fails. In these situations, ORB-SLAM is not able to recover unless there is a loop closure later on.

We therefore do not use the motion model in tracking and instead rely on brute force matching between images. To increase the chances of detecting the same features as those in the previous frame, a large number of keypoints, around 1000, are detected in each image. With so many keypoints, some might have similar appearance although they correspond to different points in the 3D environment. By using the depth information to compare patches surrounding the keypoint, DLab is able to filter out wrong matches that are only visually similar.

Brute force matching works by comparing all of the descriptors of the current image to all of those of the previous frame (here, the Hamming distance can be used since we have binary descriptors). To further ensure the quality of the matches, we use the mutually best match condition, i.e., a pair of features (q, a) with q from the current image and a from the previous image is considered a match if a is the best match for q among features in the previous image, and q is the best match for a among all features in the current image. Additionally, we apply the k-nearest neighbor algorithm to find the two closest descriptors for each query descriptor and only consider those matches with a certain distance ratio r between the nearest two:

$$d_{qa} < r \cdot d_{qb}, \tag{1}$$

where q is the query descriptor in the current image, a and b are the closest two descriptors in the previous frame, and d_{qa} as well as d_{qb} are the corresponding distances.

Similar to Endres *et al.* [11] we refine the matches using RANSAC. First the 2D points are mapped back to 3D. Then we run a number of iterations to compute the transformation between the two sets of 3D points. The outer loop in our RANSAC implementation aims at finding the transformation with the largest number of inliers. In each iteration we start out by sampling some matches, we then compute the transformation (as described below) and determine the number of initial matches that are considered as inliers using the computed transformation. The transformation is refined in an inner loop where every step uses the inliers of the previous step. The refined transformation with the smallest inlier error is kept.

We use the SVD to find the transformation, and the error is computed as the L_2 norm of the difference between a 3D point and its corresponding 3D point that has been transformed using the calculated transformation.

The inlier matches are used to find the pose estimate of the current frame. This is done by minimizing the reprojection error:

$$e = \left\| p_f - \pi(\mathbf{q}, c) \right\|, \tag{2}$$

where $\mathbf{q}$ is the 3D point in the world, c is the new camera frame, π is the projection function, and p_f is the 2D coordinate of the matched point in the previous frame. Minimizing this error is carried out by $\mathbf{g^2o}$ [23]. The new frame is added as a node with its pose initialized to that of the keyframe and each matched pair is then added as a unary edge to this node. $\mathbf{g^2o}$ optimizes this graph and yields the new pose estimate.

To prevent erroneous pose estimates, we consider the pose estimate to be incorrect when the computed translation w.r.t. the previous frame is larger than a certain threshold. We always keep the last five frames for which a transformation can be found $f_{t-1}, \ldots, f_{t-5}$. Whenever for a frame f_t no valid transformation can be found using f_{t-1}, we try to compute the transformation using the previous frames with the procedure described above. If still no valid transformation can be found, we skip the frame. This might happen when an image is too blurred and does not have descriptive features.

B. Mapping

ORB-SLAM selects a subset of images with their corresponding transformation as so-called *keyframes*. These are

then used to build a map of the environment. The selection criteria consider how many points in a local region of the map are tracked and how many features from the reference keyframe are tracked.

Accordingly, not all frames are kept as keyframes which might be a problem for loop closure as it might happen that a location that has been visited before is not recognized. We therefore always create a new keyframe from a frame with a valid transformation and rely on a post-processing step to remove keyframes that share a high number of points. In this way, graph optimization on the keyframes is still efficient and loop closures can still be reliably detected since only keyframes that are very similar to others are removed.

After creating new keyframes, the mapping part of ORB-SLAM is also responsible for performing bundle adjustment and culling redundant keyframes. These steps are carried out in a separate thread from tracking and only when there are no other keyframes waiting to be processed in the buffer. In our experiments, this led to discrepancies in the map where some regions with a high number of keyframes are not processed whereas other parts that have just a few keyframes are sent to the pipeline for bundle adjustment. In our system, therefore, tracking and mapping run in the same thread and the steps of bundle adjustment and culling are preformed every 10 keyframes.

C. Place Recognition for Loop Closing

Instead of DBoW2 [18], we use FAB-MAP [9] as the place recognition module since in our experiments we experienced several false positive loop detections. Accordingly, we modified FAB-MAP to incorporate our new binary descriptor. In the training phase of FAB-MAP, a codebook is generated from a large number of visual features. To create this vocabulary the features are clustered and representative cluster centers are used as the vocabulary words.

DLab is now used to describe keypoint patches and given that it is a binary string, two further modifications needed to be made. First, instead of the Euclidean distance to assign a feature to a cluster, we use the Hamming distance which yields the number of bits that are different between two descriptors. Second, instead of setting the cluster center to the mean of the assigned feature descriptors, we apply a majority voting scheme as in [8], i.e., a descriptor of size N contributes with N votes, where each bit in that descriptor votes for either one or zero in its corresponding position in the cluster center. After checking all the descriptors that are members of a cluster, a bit in position i in the cluster center would be set to one if it was the majority value and to zero otherwise. Whenever a new keyframe is created, we pass the keyframe to FAB-MAP and check whether a loop closure occurred. The bag-of-words descriptor of the new keyframe is added to the test set of FAB-MAP for future loop closure matches only if 10 frames have passed since the last added keyframe. This increases the chance that there is considerable visual change in the BoWs test set.

Whenever a loop closure is detected, it needs to be verified whether it is geometrically valid. Here, we use the RANSAC scheme described in Sec. IV-A to check whether a transformation exists and is supported by enough map point inliers. Fig. 4 shows the matches found after RANSAC between the current image and the image returned by FAB-MAP as the robot revisits a previous location. If the detected loop closure passes the validity test, the matched keyframe pose is optimized by minimizing the reprojection error as explained in Sec. IV-A.

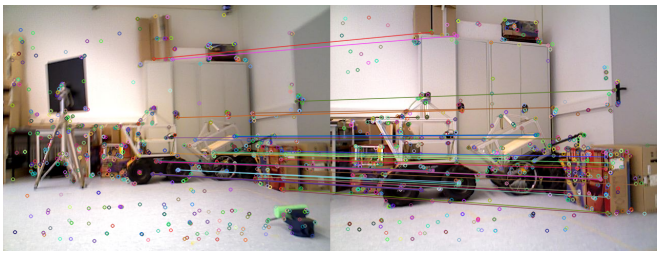

Fig. 4. The robot revisits an old place and FAB-MAP with our descriptor returns a possible candidate for loop closure. If there are enough matched features after RANSAC as in this example, a loop closure takes place.

To correct the poses of the other keyframes, we construct a graph with all keyframes between the two matched loop closure keyframes. Each node represents the camera pose and an edge is created between any two nodes that have shared map points. $\mathbf{g^2o}$ [23] is then used to optimize this graph and correct poses.

V. EXPERIMENTAL EVALUATION

In this section, we first evaluate the performance of DLab w.r.t. feature matching in comparison to existing descriptors. Then, we demonstrate the loop closure capabilities when using the new descriptor for place recognition and evaluate the absolute trajectory error during SLAM. Finally, we present experiments for a comparison of the computational costs when using the different descriptors.

A. Evaluation of Descriptors

To track frames in visual SLAM, features are matched between consecutive images and it is essential that the matches are correct to prevent a wrong pose estimate. In the following comparative experiments, we evaluate the correctness of matches using different descriptors. The evaluation follows the procedure used by Nascimento *et al.* [4] and Rublee *et al.* [1], i.e., points of interest are detected in the images of the first set with the ORB detector and a signature is generated by the descriptor. Then, a transformation (described below) is applied to create a second set of images, in which features are also detected and described. A pair of features detected in an image from the first set and the corresponding image from the second set is considered a match if the distance between the descriptors is below a certain threshold. The precision-recall measure is used to evaluate how many of the matches are correct.

Fig. 5 shows an experiment with the following setting: 1000 images are randomly selected from the TUM benchmark dataset [24] and 500 interest points are detected in each image. The images are then transformed by a translation of 10 pixels

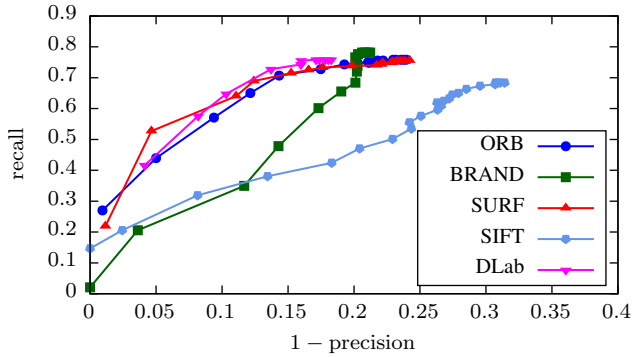

Fig. 5. Precision-Recall for various descriptors when transforming the images with a translation of 10 pixels. Each point shows the precision and recall when considering matches with a certain maximum distance between the descriptors. As can be seen, DLab performs best since it has the highest recall and precision (note that the x-axis shows 1-precision).

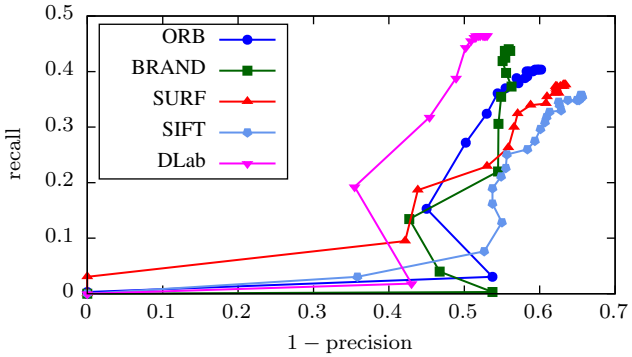

Fig. 6. Precision-recall when transforming the images with a rotation of 30°. Obviously, the precision is lower than when having a transformation with translation only. Again, DLab has the highest precision.

to the right and the detector is run on them again. The patches around the interest points are described and matched. The plot is produced by varying the threshold described earlier so every point in the curve in Fig. 5 depicts the precision and recall at a certain threshold. The range for the threshold is $[0 : 255]$ in steps of 5 for the binary descriptors ORB, BRAND, and DLab. For SIFT, the range is $[0 : 10000]$ in steps of 100, and for SURF, it is $[0 : 10]$ in steps of 0.1. As expected, increasing the threshold increases the recall and decreases precision. As can be seen, our descriptor has the best precision at high recall values.

There are two sources of error that result in wrong matches. The first error is tied to the repeatability of the detector. We noticed that it does not always detect the keypoints at exactly the same location. This naturally leads to different patches and the descriptors are consequently slightly different. The second source of error has to do with the ability of a descriptor to uniquely identify a patch. In order to reduce the number of wrong matches, we added two constraints to the matching process, namely the *mutually best match* condition and the *sufficient distance to second best match* condition (see IV-A).

In Fig. 6 we transformed the images with a rotation of 30°. As can be seen, precision is lower than in the case of a simple translation for all descriptors but still our new descriptor outperforms the others as it has the best precision at the highest recall values. We experienced that the depth cue is valuable here in differentiating features. Our descriptor produces even better results than the BRAND descriptor. As explained in Section III, splitting the results of the appearance and depth tests eliminates the ambiguity that arises in BRAND.

B. Place Recognition

Recognizing that a place has been visited before is an important component of SLAM as it helps to minimize the error in the robot's trajectory that was accumulated over time. We evaluated our modified FAB-MAP version, which works with the new binary descriptors as explained in Sec. IV-C, in comparison to DBoW2 of ORB-SLAM. Table I shows the results for two datasets recorded with our Nao robot and

TABLE I

PLACE RECOGNITION RESULTS

	Dataset 1 (2534 frames)		Dataset 2 (3160 frames)	
	Candidates	Correct	Candidates	Correct
FABMAP-DLab	1805	1805	2124	2124
FABMAP-ORB	1876	1862	2280	2266
DBoW2-DLab	2006	187	2488	428
DBoW2-ORB	2032	66	2500	125
DBoW2-ORB (original vocabulary)	2038	799	2360	430

manually labeled ground truth. The total number of frames is indicated in the top row for each trajectory. The first column shows the number of frames that the system has recognized as frames from previously seen locations, and the second column indicates how many of those candidates are correct.

For a new image, FAB-MAP returns a probability that reflects how confident it is that the image is similar to another one in its database. We follow [17] and only take into consideration those frames with a confidence value of 0.98 or higher. These are referred to as candidate frames in the second column of the table.

The results in Table I show that although using FAB-MAP with ORB leads to a higher detection rate than when using DLab, the fraction of correct detections in comparison to the total number of candidates is lower. We argue that for loop closure in SLAM, precision is more important than recall, as an incorrect loop closure might cause a wrong pose estimate from which the robot cannot recover. On the other hand, missing a loop closure is not an issue since revisiting a location generates many frames that are similar to older ones and a loop closure that is not detected by one frame will be detected by another with a high chance.

The original implementation of ORB-SLAM used DBoW2 as its place recognition component. We ran three experiments with DBoW2; the first two use the same pool of images to generate the vocabulary as with the FABMAP experiments, and the last experiment uses the vocabulary made publicly available by the authors of DBoW2. In all these experiments,

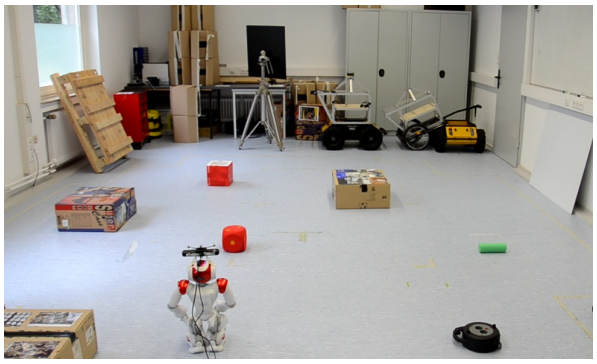

Fig. 7. The environment where the trajectories were recorded.

TABLE II

ABSOLUTE TRAJECTORY ERROR

Datasets	Nao1	Nao2	Nao3	Nao4	Pioneer
Our approach (in m)	0.08	0.1	0.04	0.05	0.03
ORB-SLAM (in m)	0.19	0.23	0.04	0.09	–

DBoW2 generated a large number of false positives, as shown in Table I, which motivated us to use FABMAP instead. Note that DBoW2 excludes similar images that were already known to share map points to restrict candidates to those taken at different time intervals, accordingly the absolute number of candidates and correct matches is naturally reduced.

C. Simultaneous Localization and Mapping

Furthermore, we evaluated the complete SLAM system with our new descriptor. We used the evaluation tool provided by the TUM benchmark utility [24] to compute the absolute trajectory error (ATE):

$$\text{ATE} = \sqrt{\frac{\sum_N \|X_t - G_t\|^2}{N}}, \tag{3}$$

where X_t and G_t are the estimated and ground truth poses respectively at time t, and N is the number of pose estimates.

We recorded several trajectories using a Nao robot with an ASUS Xtion camera mounted on top (see Fig. 1) in an environment with a motion capture system that provided the ground truth data. Fig. 7 shows the environment in which we performed the experiments. Table II shows the corresponding ATEs.

The top two rows of Fig. 8 show that our system performs well with data recorded with the Nao robot and outperforms ORB-SLAM (original version). Pose tracking with legged robots such as the Nao is more challenging than with wheeled robots since the movement of the head can result in blurry images that may not have well-defined features. To check whether our system also performs well in a different setting, we tested our approach on a dataset from the TUM benchmark datasets [24]. The last row of Fig. 8 shows the estimated and the ground truth trajectory for a Pioneer wheeled robot. In contrast to ORB-SLAM which is not able to track the robots pose, our system accurately tracks the trajectory and performs loop closing.

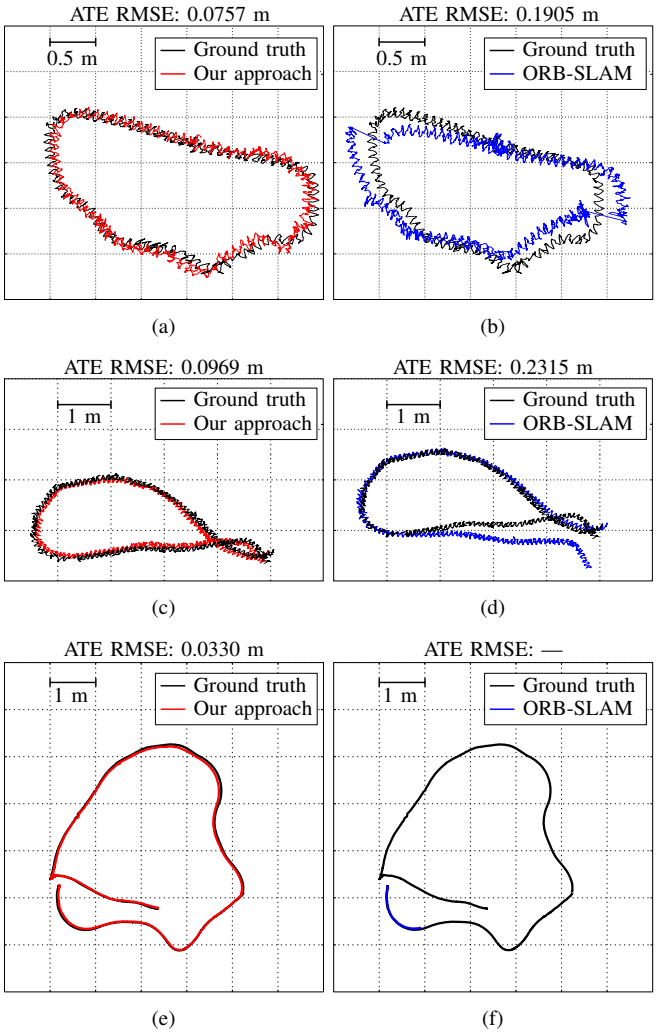

Fig. 8. SLAM results on several datasets. Our system (left column) performs well and the ATE is lower than when using ORB-SLAM (right column). The top two rows show the results for data recorded with the Nao robot in the environment shown in Fig. 7 (datasets Nao1 and Nao2 in Table II) whereas the last row shows the results for data of a Pioneer robot.

TABLE III

TIME TO COMPUTE 100 DESCRIPTORS OF EACH TYPE

Descriptor	ORB	DLab	SURF	BRAND	SIFT
Time (ms)	4	14	94	125	132

As can be seen from Fig. 8 (f), ORB-SLAM cannot track the pose as the robot makes a turn. ORB-SLAM first uses the motion model from the previous frames to initialize the new pose. This new pose, however, is not supported by enough matches so it then tries to compute the transformation from the keyframe of the local map. Since there was a rotation in the movement of the robot, the reference keyframe no longer shares enough points with the current frame and matching fails.

D. Computational Cost

Finally, Table III shows a comparison w.r.t. computation time on a single Intel Core Pentium 987 CPU. Here, we

evaluated the time needed to create 100 descriptors of each type. ORB is the only descriptor that is faster than ours. However, it only carries appearance information. DLab is significantly faster than BRAND which also considers depth information. The main speed-up is attributed to how the depth information is used. We compare depth values directly, whereas BRAND computes normals from the point clouds and uses them in the geometric tests. Furthermore, as shown in the evaluation in Sec. V-A, our descriptor outperforms BRAND in feature matching.

A more detailed breakdown of the runtime of our descriptor showed that the most expensive step is the computation of the patch orientation. It uses the same method as in SURF where the Haar wavelet responses are computed and summed in the x and y directions. Note that the runtimes refer to single core computation. We expect faster computation times with parallelization.

VI. CONCLUSION

In this work, we introduced DLab, a new binary descriptor that uses the intensity, depth, and color information. For SLAM with humanoids, we proposed to apply DLab within ORB-SLAM, using a modification of FAB-MAP with our new binary descriptor for place recognition.

We thoroughly evaluated our approach. First, we conducted experiments comparing the performance of DLab to existing descriptors. We verified that the new descriptor has a higher precision with comparable recall. This makes it especially useful for SLAM applications where it is hard to recover from incorrect data associations as they can lead to wrong pose estimates. Furthermore, we evaluated the place recognition capabilities with DLab. We found that although it returns a smaller number of similar images, the percentage of the correct candidates returned is higher when compared to other approaches, which is desirable for SLAM since a robot may never recover from a wrong loop closure. Finally, we evaluated our complete visual SLAM system on various real-robot data sets. The results obtained with a Nao robot equipped with an RGB-D camera demonstrate that our system outperforms ORB-SLAM as it follows the ground truth trajectory more closely and has a lower absolute trajectory error. DLab is able to reliably track features even in sequences with blurred images, which frequently occur with walking humanoids, thereby allowing for more robust navigation in complex environments.

REFERENCES

[1] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2011.

[2] R. Hänsch, T. Weber, and O. Hellwich, "Comparison of 3D interest point detectors and descriptors for point cloud fusion," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, no. 3, 2014.

[3] F. Tombari, S. Salti, and L. Di Stefano, "A combined texture-shape descriptor for enhanced 3D feature matching," in *Proc. of the IEEE Intl. Conf. on Image Processing (ICIP)*, 2011.

[4] E. Nascimento, G. Oliveira, M. Campos, A. Vieira, and W. Schwartz, "BRAND: A robust appearance and depth descriptor for RGB-D images," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[5] X. Xiao, S. He, Y. Guo, M. Lu, and J. Zhang, "Bag: A binary descriptor for rgb-d images combining appearance and geometric cues," in *International Conference on Cognitive Systems and Signal Processing*. Springer, 2016, pp. 65–73.

[6] C. Romero-González, J. Martínez-Gómez, I. García-Varea, and L. Rodríguez-Ruiz, "Binary patterns for shape description in rgb-d object registration," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 2016, pp. 1–9.

[7] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Trans. on Robotics (TRO)*, vol. 33, no. 5, 2017.

[8] D. Gálvez-López and J. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. on Robotics (TRO)*, vol. 28, no. 5, 2012.

[9] A. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, and G. Wyeth, "OpenFABMAP: An open source toolbox for appearance-based loop closure detection," in *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2012.

[10] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments," *Intl. Journal of Robotics Research (IJRR)*, vol. 31, no. 5, 2012.

[11] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2012.

[12] R. Mur-Artal, J. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. on Robotics (TRO)*, vol. 31, no. 5, 2015.

[13] N. Figueroa, H. Dong, and A. El Saddik, "A combined approach toward consistent reconstructions of indoor spaces based on 6D RGB-D odometry and KinectFusion," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 2, 2015.

[14] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. of the Intl. Symp. on Mixed and Augmented Reality (ISMAR)*, 2011.

[15] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[16] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended KinectFusion," in *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, 2012.

[17] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *Intl. Journal of Robotics Research (IJRR)*, vol. 27, no. 6, 2008.

[18] R. Mur-Artal and J. D. Tardós, "Fast relocalisation and loop closing in keyframe-based SLAM," in *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2014.

[19] N. Suenderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free," in *Proc. of Robotics: Science and Systems (RSS)*, 2015.

[20] O. Stasse, A. Davison, R. Sellaouti, and K. Yokoi, "Real-time 3D SLAM for humanoid robot considering pattern generator information," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[21] A. Pretto, E. Menegatti, M. Bennewitz, W. Burgard, and E. Pagello, "A visual odometry framework robust to motion blur," in *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2009.

[22] G. Oriolo, A. Paolillo, L. Rosa, and M. Vendittelli, "Humanoid odometric localization integrating kinematic, inertial and visual information," *Autonomous Robots*, vol. 40, no. 5, 2016.

[23] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g^2o: A general framework for graph optimization," in *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2011.

[24] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012.