DIIS - I3A
C/ María de Luna num. 1
E-50018 Zaragoza
Spain

Internal Report: 2007-V08

SURF features for efficient robot localization with omnidirectional images

A. C. Murillo, J. J. Guerrero and C. Sagüés

If you want to cite this report, please use the following reference instead:
SURF features for efficient robot localization with omnidirectional images, A. C. Murillo, J. J. Guerrero and C. Sagüés, *IEEE Int. Conference on Robotics and Automation*, pages 3901-3907, Rome - Italy, April 2007.

This work was supported by projects DPI2006-07928, IST-1-045062-URUS-STP.

SURF features for efficient robot localization with omnidirectional images

A. C. Murillo, J. J. Guerrero and C. Sagüés

***Abstract*— Many robotic applications work with visual reference maps, which usually consist of sets of more or less organized images. In these applications, there is a compromise between the density of reference data stored and the capacity to identify later the robot localization, when it is not exactly in the same position as one of the reference views. Here we propose the use of a recently developed feature, SURF, to improve the performance of appearance-based localization methods that perform image retrieval in large data sets. This feature is integrated with a vision-based algorithm that allows both topological and metric localization using omnidirectional images in a hierarchical approach. It uses Pyramidal kernels for the topological localization and three-view geometric constraints for the metric one. Experiments with several omnidirectional images sets are shown, including comparisons with other typically used features (radial lines and SIFT). The advantages of this approach are proved, showing the use of SURF as the best compromise between efficiency and accuracy in the results.**

I. INTRODUCTION

Often mobile robots have reference maps at their disposal or are at least able to construct their own. Working with vision sensors, these maps usually are a more or less organized set of images, frequently grouped in clusters corresponding to different locations or nodes, e.g rooms. The robot localization needs to be more or less accurate depending on the task to perform afterwards. For instance, topological localization is less accurate but faster and more useful to communicate with humans. However, for navigation or interaction with objects (e.g. to avoid them or to pick them) metric information is needed. In earlier work [1], we have presented an appearance-based localization method that uses a hierarchical approach to obtain topological and metric localization information from omnidirectional images.

Omnidirectional vision and hierarchical localization are two topics of interest nowadays. Omnidirectional vision has become widespread in the last years, and has many well-known advantages as well as extra difficulties compared to conventional images. There are many works using all kind of omnidirectional images, e.g. a map-based navigation with images from conic mirrors [2] or localization based on panoramic cylindric images composed of mosaics of conventional ones [3]. Hierarchical localization processes have been also a field of study in the previous years, e.g., [4], [5], [6]. Usually their goal is to localize the robot as fast as possible with a lot of reference information, then they perform different steps pruning the reference data in each one to save time. Other option to improve the efficiency with big amounts of data consists of using efficient data structures that allow us to speed up the computations, as in [7], using trees to increase the efficiency with a lot of data in simultaneous localization and mapping, or in [8], using clusters of features and trees to efficiently search in a very big image database.

This work explains how to obtain an efficient global localization combining SURF features with a hierarchical method, which provides topological and metric information with regard to a big set of reference images or visual memory. Here the automatic construction of this reference set, or topological map, is not studied, but there are many recent works dealing with this problem, such as [9] or [10]. The subjects of our work are open issues of hierarchical localization methods: improving the accuracy in the final localization and increasing the speed to deal with big reference data sets. Here, we improve the efficiency and robustness of the work done in [1]. There, thanks to the use of three-view geometric constraints, accurate metric localization information can be obtained from a minimal set of reference views. Its localization accuracy depends only on the wide baseline it is able to deal with the image feature used, while in other methods for image based metric localization, e.g. the one proposed in [5], the accuracy depends on the separation between reference images in the image grid stored. The improvements here with regard to [1] are mostly due to the integration with a recently developed local feature named Speeded-Up Robust Features (SURF) [11]. This feature allows us to better cope with wide baseline situations in a efficient way.

This paper introduces in the field of vision based localization the usage of SURF. It has been previously used, e.g, for object recognition in a museum guide application [12]. However, it had not yet been applied in the robotics field nor in omnidirectional images and seems convenient in these tasks. It states to have more discriminative power than other state-of-the-art features such as SIFT [13], yet can be computed more efficiently and yields a lower dimensional feature descriptor resulting in faster matching. The construction of the SURF features is quite convenient also for hierarchical approaches. Initially, for the first rough steps of the hierarchy, a faster and smaller feature descriptor vector can be extracted. Later, for the more accurate steps of the process, a more accurate descriptor vector can be obtained. The experimental section compares the performance of SURF against the popular SIFT (version provided by D. Lowe in [13]), the most popular wide-baseline feature in the

This work was supported by the projects DPI2003-07986, DPI2006-07928 and IST-1-045062-URUS-STP.

A. C. Murillo, J. J. Guerrero and C. Sagüés are with DIIS - I3A, University of Zaragoza, Spain. acm@unizar.es

last years, and against radial lines, a simple, fast and easy feature to extract in omnidirectional images [1]. The results of our experiments, with two different data sets, show that the best compromise between performance and efficiency is obtained with SURF.

We provide more details about SURF in section II, while the localization process used is detailed in section III. Finally in section IV, exhaustive experiments with different omnidirectional image sets are shown to validate the proposal.

II. Speeded Up Robust Features (SURF)

SURF is a local feature recently presented in [11]. This section shows a brief summary of its construction process, first the interesting point localization and after the feature descriptors computation.

A. Interest Point Localization.

The SURF detector is based on the Hessian matrix. Given a point $\mathbf{x} = [x, y]$ in an image I, the Hessian matrix $H(\mathbf{x}, \sigma)$ in x at scale σ is defined as follows

$$H(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix}, \quad (1)$$

where $L_{xx}(\mathbf{x}, \sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial}{\partial^2 x} g(\sigma)$ with the image I in point **x**, and similarly for $L_{xy}(\mathbf{x}, \sigma)$ and $L_{yy}(\mathbf{x}, \sigma)$. In contrast to SIFT, which approximates Laplacian of Gaussian (LoG) with Difference of Gaussians (DoG), SURF approximates second order Gaussian derivatives with box filters. See an example of one of this filters for the lowest scale analyzed in Fig. 1 Image convolutions with these box filters can be computed rapidly by using integral images [14].

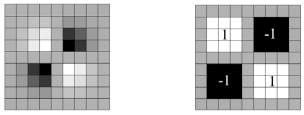


Fig. 1. Left: gaussian second order derivative in xy-direction. Right: corresponding box filter approximation.

The location and scale of interest points are selected by relying on the determinant of the Hessian. Interest points are localized in scale and image space by applying a non-maximum suppression in a 3 x 3 x 3 neighbourhood. Finally, the local maxima found of the approximated Hessian matrix determinant are interpolated in scale and image space. For more details, see [11].

B. Interest Point Descriptor.

In a first step, SURF constructs a circular region around the detected interest points in order to assign a unique orientation to the former and thus gain invariance to image rotations. The orientation is computed using Haar wavelet responses in both x and y directions. The Haar wavelets can be quickly computed via integral images, similar to the Gaussian second order approximated box filters. The dominant orientation is estimated and included in the interest point information.

In a next step, SURF descriptors are constructed by extracting square regions around the interest points. These are oriented in the directions assigned in the previous step. The windows are split up in 4 x 4 sub-regions in order to retain some spatial information. In each sub-region, Haar wavelets are extracted at regularly spaced sample points. The wavelet responses in horizontal and vertical directions (dx and dy) are summed up over each sub-region. Furthermore, the absolute values $|d_x|$ and $|d_y|$ are summed in order to obtain information about the polarity of the image intensity changes. Hence, the underlying intensity pattern of each sub-region is described by a vector $\mathbf{V} = [\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|]$. The resulting descriptor vector for all 4 x 4 sub-regions is of length 64, giving the standard SURF descriptor, SURF-64. It is possible to use 3 x 3 sub-regions instead, then we obtain a shorter version of the descriptor, SURF-36, that will be also used in our applications. Notice that the Haar wavelets are invariant to illumination bias and additional invariance to contrast is achieved by normalizing the descriptor vector to unit length.

An important characteristic of SURF is the fast extraction process, that takes profit of integral images and a fast non-maximum suppression algorithm. Also is very convenient the fast matching speed it permits, mainly achieved by a single step added to the indexing based on the sign of the Laplacian (trace of the Hessian matrix) of the interest point. The sign of the Laplacian distinguishes bright blobs on a dark background from the inverse situation. Bright interest points are only matched against other bright interest points and similarly for the dark ones. This minimal information permits to almost double the matching speed and it comes at no computational costs, as it has already been computed in the interest point detection step.

III. Efficient vision based localization

This section explains a hierarchical method to efficiently localize the actual position of the robot with regard to a big set of reference images or visual memory (VM).

A. Similarity evaluation for topological localization

This part details the process for the topological localization, i.e., to recognize the room, which evaluates the similarity between the current view and the images from the VM.

First, a color global image descriptor is applied as a pre-filtering for the reference views, as described in [1], rejecting those images with very low similarity in this descriptor. This filter can not be very strict, as the global descriptors are very sensitive to occlusions and noise, but it is very useful to reduce the set of candidate locations for the next steps.

The rest and more important part of the similarity evaluation assigns a more accurate similarity value to each reference view that passed the initial pre-filter. This has been done with two different methods, one based on a Pyramidal matching and other based on a nearest neighbour (NN) matching. In general, the first one is more efficient and robust, but this is not true for long descriptor vectors.

Therefore, SIFT descriptor seems not suitable for this method due to its descriptor size (128), then we tried also the second similarity measurement to make our experimental validation more complete.

1) Similarity based on Pyramidal matching: We use a similarity evaluation process based on the Pyramid matching kernels proposed in [15]. It allows local feature matching between the reference images and the current one with linear cost in the number of features. It takes into account the distribution of the local features, not only their descriptors. The features descriptors vectors are used to implement this mentioned matching structures. The idea consists of building for each image several multi-dimensional histograms (one dimension per descriptor), where each feature falls in one of the histogram bins. Each descriptor value is rounded to the histogram resolution, which gives a set of coordinates that indicates the bin corresponding to that feature.

Several levels of histograms are defined. In each level, the size of the bins is increased by powers of two until all the features fall into one bin. The histograms of each image are stored in a vector (or pyramid) ψ with different levels of resolution. The similarity between two images, the current (c) and one of the VM (v), is obtained by finding the intersection of their corresponding pyramids of histograms

$$S(\psi(c), \psi(v)) = \sum_{i=0}^{L} w_i N_i(c, v) \ , \tag{2}$$

with N_i the number of matches (features that fall in the same bin of the histograms, see Fig. 2) between images c and v in level i of the pyramid. w_i is the weight for the matches in that level, that is the inverse of the current bin size (2^i). This distance is divided by a factor determined by the self-similarity score of each image, in order to avoid giving advantage to images with bigger sets of features, so the distance obtained is

$$S_{cv} = \frac{S(\psi(c), \psi(v))}{\sqrt{S(\psi(c), \psi(c))\ S(\psi(v), \psi(v))}} \ . \tag{3}$$

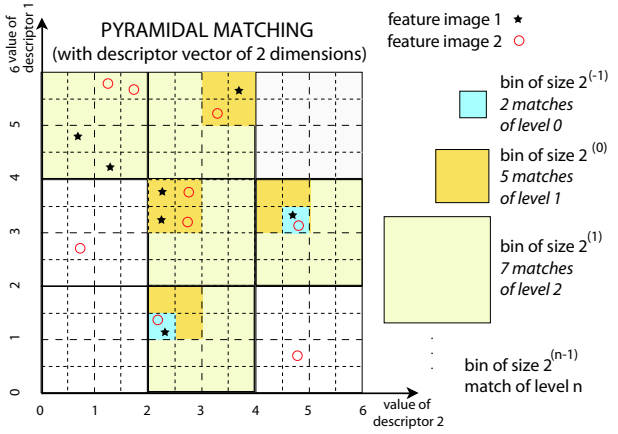


Fig. 2. Example of Pyramidal Matching, with correspondences in level 0, 1 and 2. For graphic simplification, with a descriptor of 2 dimensions.

Note that the matching obtained does not have all matches feature-to-feature, it happens often, specially when using bigger bin-sizes, that more than one feature from each image falls in a certain histogram cell (as happens in Fig. 2), so we count two matches there but we can not distinguish them.

2) Similarity based on Nearest Neighbour matching: We can compute a similarity score that depends on the matches found (n) between the pair of images, weighted by the average distance (d) between the features matched. It has to take also into account the number of features not matched in each image (F_1 and F_2 respectively) weighted by the probability of occlusion of the features (P_o). The defined dissimilarity (DIS) measure is

$$DIS = n\ d + F_1(1 - P_o) + F_2(1 - P_o) \ . \tag{4}$$

Once the most similar image from the VM to the current one is determined with one of the previously explained similarity evaluations, the annotations of this chosen image indicate the room where the robot is currently.

B. Metric localization through the Radial Trifocal Tensor

For many applications, a localization information more accurate than the current room is needed. The structure and motion parameters have been typically recovered in computer vision applications from geometric constructions such as the fundamental matrix, with well known structure from motion algorithms [16]. The multi-view geometry constraint for three 1D views is the 1D trifocal tensor [17]. In case of omnidirectional images, accurate robot and landmarks localization can be achieved from the 1D radial trifocal tensor [18]. This tensor is robustly estimated from trios of correspondences, applying a robust method (*ransac*) in the three-view matching process to simultaneously reject outliers and estimate the tensor.

In our case, the three omnidirectional images used are the current one and two from the reference database (the most similar found and one neighbour). The image feature dimension used is their orientation ϕ, relative to the direction where the camera is pointing (see Fig. 3). It must be expressed as 1D homogeneous coordinates $\mathbf{r} = [\sin\phi, \cos\phi]$.

The projections of a certain feature $\mathbf{v}$ in the three views $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ are constrained by the trilinear constraint imposed by the 1D trifocal tensor

$$\sum_{i=1}^{2} \sum_{j=1}^{2} \sum_{k=1}^{2} T_{ijk}\, \mathbf{r}_{1(i)}\, \mathbf{r}_{2(j)} \mathbf{r}_{3(k)} = 0, \tag{5}$$

where T_{ijk} $(i, j, k = 1, 2)$ are the eight elements of the $2 \times 2 \times 2$ trifocal tensor and subindex $_{(\cdot)}$ are the components of vectors $\mathbf{r}$.

The 1D tensor can be estimated with five matches and two additional constraints [19] defined for the calibrated situation (internal parameters of the camera are known). From this tensor estimated for omnidirectional images, without other camera calibration than the center of projection, a robust set of matches, the camera motion and the structure of the scene can be computed in a closed form [18]. Fig. 3 shows a feature projected in three views and the location parameters estimated.

Using 1D bearing-only data, it is well known that three views are needed to recover the structure of the scene.

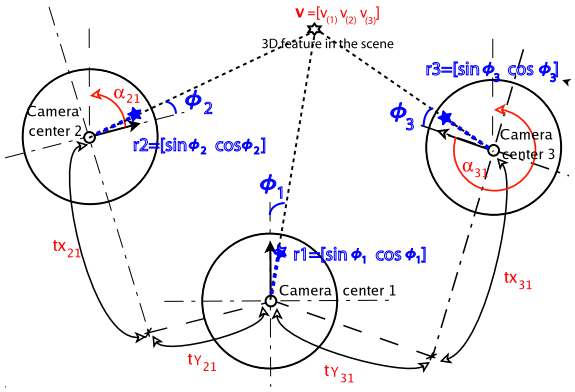


Fig. 3. Landmark projection in 1D views and motion parameters estimated: translation directions $[t_{x21}, t_{y21}]$, $[t_{x31}, t_{y31}]$ and rotations α_{21}, α_{31}.

But there are more advantages in computing the metric localization with this three view geometry based approach. First, as it uses two reference images and we suppose an annotated reference set, the reference information between these two views helps to solve the ambiguity and the scale of the localization obtained from the 1D tensor. Second, using three views makes the matching more robust without too much computing overload (matches between images in the VM can be pre-computed and stored). The fact of using only the angular coordinate also helps to the robustness, as in omnidirectional images it is more accurate than the other polar coordinate (the radial coordinate). Finally, it can also help to automatically detect situations of failure. For example, if the two reference images that obtained the highest similarity scores are not from the same room in the database, it can give us indications of some mistake and allow us to act accordingly.

IV. Localization Experiments

This section shows the performance of the method explained in previous sections for the topological localization (i.e. current room recognition) and for the metric localization.

The results obtained with the new feature SURF [11] were compared to results with the most commonly used wide-baseline feature, SIFT [13], and to results with radial lines, as they are simple and fast features previously used for these tasks. The feature extraction was performed with the implementation provided in the given references. The radial line matching was performed as in [1] and the matching of SURF and SIFT was a typical nearest neighbour algorithm that considers a match correct if the distance between first ($d1$) and second ($d2$) nearest neighbour fits $d1 \leq threshold * d2$.

Two data sets of omnidirectional images were used, *Almere* (standard data set provided in [20]) and our own (named data set *LV*). We decided to use also this second data set because ground truth data was available for its images, which was convenient to measure the errors in the localization. This visual memory has 70 omnidirectional images (640x480 pixels). 37 of them are sorted, classified in four different rooms, with between 6 and 15 images of each one (depending on the size of the room). The rest corresponds to unclassified ones from other rooms, buildings or outdoors. From the *Almere* data set, we have extracted the frames from the low quality videos provided from the rounds 1 and 4 (2000 frames extracted in the first, and 2040 in the second). We kept just every 5th frame. From these, we assigned half for the visual memory (the odd frames: 0-10-20-30- ...) and the other half for testing (5-15-25-...). The images correspond to a robot-tour around a typical house environment with several rooms (living-room, kitchen,...). Fig. 4 shows a scheme of both databases, in case of data set *LV* with details of the relative displacements between views. All images have been acquired with an omnidirectional vision sensor with hyperbolic mirror.

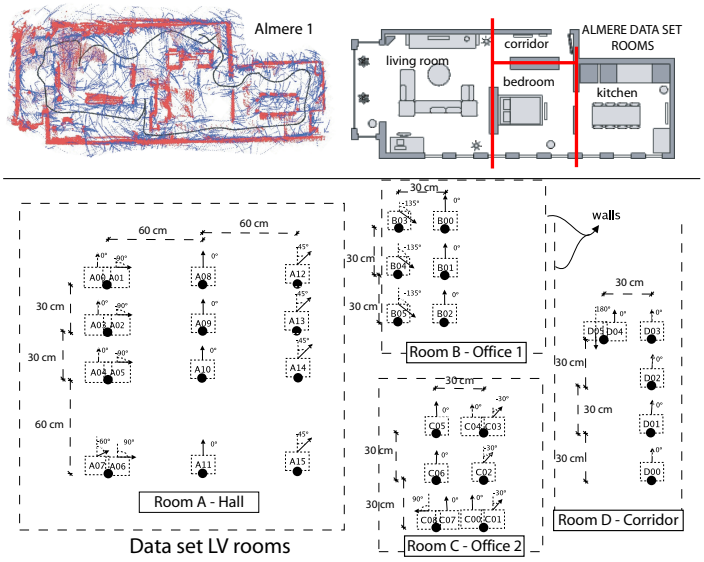


Fig. 4. Grids of images in rooms used in the experiments. Top: *Almere* data set (typical home environment). Bottom: data set *LV* (typical office environment).

A. Topological localization performance

This section shows the topological localization performance of the methods explained in section III-A. The experiments consisted of running the similarity evaluation to localize the room where the robot is. Three different cases were studied:

- Case 1: data set *LV*. This data set contains views that were taken separately in each room (not during a robot tour), taking annotations for the ground truth. In this case, the localization was done using a view from data set *LV* as query and the other views from the same data set as VM.

- Case 2: *Almere1*$\propto$*1*. In this case, the query image to be localized belongs to Almere-data set *round 1*, and the VM was composed with other images from the same round. Images in this *round 1* were taken during a robot tour around the environment shown in Fig.4.

- Case 3: *Almere4*$\propto$*1*. This is the most complicated case, because the VM was built from Almere *round 1* images but the query ones belonged to *round 4*. This tour was done in the same environment but with many occlusions and noise.

The results of the room recognition performed in these cases are show in Table I. The first row, *pre-filter*, gives a summary of the performance of the pre-filtering used in the similarity evaluation, showing the average number of images rejected ($rej.$) and the false negatives wrongly rejected ($f.n.$).

Column $1Ok$ indicates the percentage of tests where the most similar image found was correct. The time information in column T/T_{surf} is just a comparative of the relative speed to compare a query with the reference set of images for each of the features used. The surf execution time (T_{surf}) is taken as reference and the others are relative to it in each case. Note that the implementations were run in Matlab and were not optimized for speed.

The results for radial lines were definitely better with the Pyramidal matching classification, as the correctness with the NN evaluation was similar, but the execution time was smaller for the Pyramidal matching (around 25% less). However, note that when the difficulty of the case studied increases, the lines performance decreases more than with the other features. The results for SIFT shown in Table I were obtained with the NN similarity classification, because the ones obtained with the Pyramidal matching were worse, e.g., just 60% correct classifications $(1Ok)$ using data set *LV* while with NN it achieved a 89%. This and specially the high execution times confirmed that the Pyramidal matching is not suitable for SIFT (it took around 30 times more than the Pyramidal matching with SURF-36). This was already expected because of the big size of its descriptor vector. Yet, the performance of SIFT using the NN similarity (in Table I) was lower than the obtained with SURF-36 features and the Pyramidal matching. To sum up, the best compromise between correctness and execution time was obtained using SURF with the Pyramidal matching classification.

The case 3, Almere4$\propto$1, was the most difficult and indeed the one with worse performance. However, analyzing some of the tests that failed, the results were not completely wrong, i.e., many of these tests used views close to an open door to next room, therefore many features matched were from that next room already visible in the image. This made the algorithm give another room as localization. It could be studied in the future a more complex matching process which takes into account that different parts of the image can belong to different rooms, then these situations could be handled.

TABLE I
ROOM RECOGNITION: PRE-FILTERING AND SIMILARITY EVALUATION RESULTS.

data set	*LV*		*Almere1*$\propto$*1*		*Almere4*$\propto$*1*	
pre-filter	$rej.$	$f.n.$	$rej.$	$f.n.$	$rej.$	$f.n.$
	60	3	18.4	4.5	19.5	5.6
	1 Ok	T/T_{surf}	*1 Ok*	T/T_{surf}	*1 Ok*	T/T_{surf}
lines-22	89%	0.1	73%	0.2	47%	0.2
surf-36	97%	1	95%	1	67%	1
sift*-128	89%	3	80%	10	60%	10

The number after each feature type shows the length of its descriptor set.
* Results with SIFT using NN similarity evaluation, results with the other features using the Pyramidal one.

With regard to robustness, we can consider this topological localization approach good, as we have tried to reduce the size of the reference images to half and the performance stayed similar to the shown results. Reducing the reference image set is not a problem for the correctness in the topological localization (to identify the current room). Next section results show that the minimal amount required of reference images is set by the ability of the features used to obtain three view matches in widely separated images. Not all the features allow us to reduce in the same amount the density of the reference data, due to the different performance of each feature for wide-baseline matching.

B. Metric localization performance

Metric localization tests were performed with randomly chosen samples from the available data sets. A query image was picked, its most similar was detected using the previously explained similarity evaluation, and with those two and one neighbouring in the VM, we performed a robust three-view matching and tensor estimation to recover the camera and landmarks location. Not only the metric errors should be taken into account, but also the performance with more or less separated views. The discrepancy between images that we are able to deal during the matching indicates with the density of images needed in the VM.

First, two representative tests are detailed in Table II. They are both using data set *LV* because there was ground truth available only for that data. The errors obtained were good, specially taking into account the accuracy of the ground truth, that was manually obtained measuring with metric tape and goniometer. The description of each test is as follows:

- Test 1. Typical trio of images obtained after evaluating the similarity of a query. In this case, the three features where robust enough to provide matches to estimate correctly the 1D radial tensor (we see acceptable errors for three of them and good matching results in Fig.5).

- Test 2. This is a more difficult case, where we got almost the minimum necessary matches to estimate the tensor. Note that the method is still working properly. The worse performance of SIFT and lines in this case is explained by the few three-view matches obtained, while SURF obtained a little bigger set, enough to make the geometry estimation more accurate. Fig. 6 shows SURF matches. A more advanced matching process could help to increase the set of matches and get better performance with all features.

TABLE II
ROBOT METRIC LOCALIZATION ERRORS ESTIMATING THE 1D TENSOR WITH DIFFERENT FEATURES (AVERAGE FROM 20 EXECUTIONS).

	TEST 1-A10-A08-A09				TEST 2-D00-D02-D05			
Localization	α_{21}	α_{31}	t_{21}	t_{31}	α_{21}	α_{31}	t_{21}	t_{31}
lines-22	1.4	1.2	0.9	0.6	2	3.5	7	3.4
surf-64	1.2	0.9	0.9	0.4	1.6	0.4	2.4	4.6
sift-128	1.3	0.9	1	0.3	1.8	2.7	6.6	11

Average errors (degrees) in 20 executions for rotations α and directions of translation $\mathbf{t} = [t_x, t_y]$ (see these parameters in Fig. 3).
The number after each feature type shows the length of its descriptor set.

Several tests with the *Almere* data set were performed too, to evaluate the matching with more challenging views. We had no ground truth to compare the localization results obtained there, so no errors are measured here. The experiments performed with this data set can be grouped in two tests:

- Test 3. *Almere1*$\propto$*1*. Different random queries from Almere data set round 1 were compared against the VM built also from that round (indeed, different test and training views). The matching results were similar to Test 1 results.

- Test 4. *Almere4*$\propto$*1*. This was the most challenging case. Random query images from Almere data set round 4 (highly occluded) were compared against the VM built from Almere round 1. Fig. 7 is an example of SURF matching results for this test. We can see in the same figure the location parameters obtained, that were stable after several execution, and we can not measure the errors, but they seem acceptable. The results from SIFT were similar to SURF, but lines were not able to obtain enough three view matches, showing that they can not deal with cases where the baseline increases significantly. In general, once the most similar from the VM to the query was found, if the neighbouring image selected was further than 10 or 20 frames, the lines started to behave bad, while for SIFT and SURF 50 frames and higher distance was still ok.

In one hand, for the simpler cases, all features performed similarly well with regard to the matching and localization. Here, the radial lines had the advantage of being faster in extraction and matching. Although they got fewer matches, the radial lines usually represent useful features in the scene, such as walls, doors,... On the other hand, using more separated views SURF and SIFT performance was better. As shown in Test 4. In this more challenging cases, notice the advantages of SURF, which is faster than SIFT getting similar accuracy in the localization parameters. The average time for SURF three view matching was three times less than for SIFT (using the same matching method for both), due to the shorter SURF descriptor vector. Moreover, in our experiments SIFT extraction was almost three times slower than SURF's.

Taking into account both topological and metric localizations results, we can conclude the better performance of SURF, as it is always the best performing or in case of similar accuracy is much faster than the other options.

V. Conclusion

In this work we have presented an appearance based hierarchical method to localize a robot against a visual memory (VM) of reference omnidirectional images. The proposal combines the use of a recently developed feature (SURF) in two efficient steps. First using Pyramidal matching kernels to evaluate fast the similarity with the VM and to obtain a topological localization. Secondly, using the most similar image found in the VM, a metric localization is computed. That is made from a 1D trifocal tensor robustly estimated from three view feature matches. One of the big advantages using the proposed method is that we can get accurate metric localization even if the reference image set has low density. SURF features have been extensively compared against radial lines and SIFT features, showing SURF the best compromise between efficiency and accuracy in all the process, giving accurate results and allowing faster computations.

VI. Acknowledgments

Thanks to H. Bay for his helpful comments in this work.

References

[1] A. C. Murillo, C. Sagüés, J. J. Guerrero, T. Goedemé, T. Tuytelaars, and L. Van Gool. From omnidirectional images to hierarchical localization. *Robotics and Autonomous Systems*, 2007. In press.

[2] Y. Yagi, Y. Nishizawa, and M. Yachida. Map-based navigation for a mobile robot with omnidirectional image sensor copis. *IEEE Trans. Robotics and Automation*, 11(5):634–648, 1995.

[3] Chu-Song Chen, Wen-Teng Hsieh, and Jiun-Hung Chen. Panoramic appearance-based recognition of video contents using matching graphs. *IEEE Trans. on Systems Man and Cybernetics, Part B*, 34(1):179–199, 2004.

[4] J. Gaspar, N. Winters, and J. Santos-Victor. Vision-based navigation and environmental representations with an omnidirectional camera. *IEEE Trans. on Robotics and Automation*, 16(6):890–898, 2000.

[5] E. Menegatti, T. Maeda, and H. Ishiguro. Image-based memory for robot navigation using properties of the omnidirectional images. *Robotics and Autonomous Systems*, 47(4):251–267, 2004.

[6] J. Košecká, F. Li, and X. Yang. Global localization and relative positioning based on scale-invariant keypoints. *Robotics and Autonomous Systems*, 52(1):27–38, 2005.

[7] U. Frese and L. Schröder. Closing a million-landmarks loop. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 5032–5039, 2006.

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

[9] T. Goedemé, T. Tuytelaars, and L. Van Gool. Visual topological map building in self-similar environments. In *Int. Conf. on Informatics in Control, Automation and Robotics*, pages 3–9, 2006.

[10] Z.Zivkovic, B.Bakker, and B.Krose. Hierarchical map building using visual landmarks and geometric constraints. In *In Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 7–12, 2005.

[11] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *The ninth European Conference on Computer Vision*, 2006, http://www.vision.ee.ethz.ch/ surf/.

[12] Herbert Bay, Beat Fasel, and Luc Van Gool. Interactive museum guide: Fast and robust recognition of museum objects. In *First international workshop on mobile vision*, 2006.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004, http://www.cs.ubc.ca/ lowe/keypoints/.

[14] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 511–518, 2001.

[15] K. Grauman and T. Darrell. The pyramid match kernels: Discriminative classification with sets of image features. In *IEEE Int. Conf. on Computer Vision*, pages 1458–1465, 2005.

[16] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, Cambridge, 2000.

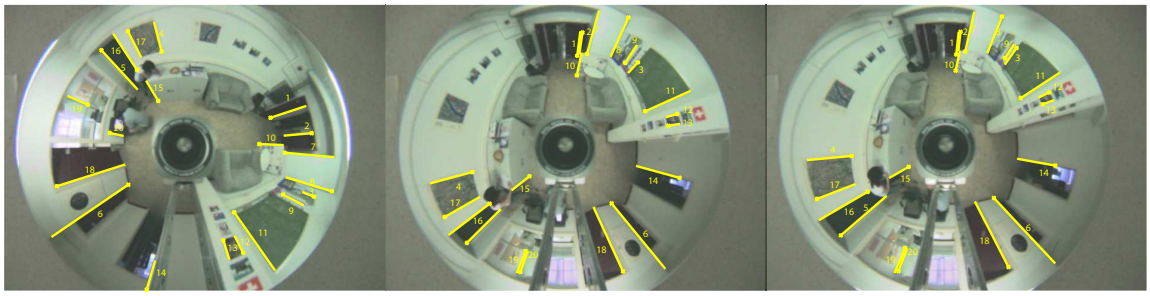
[17] O. Faugeras, L. Quan, and P. Sturm. Self-calibration of a 1d projective camera and its application to the self-calibration of a 2d projective camera. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(10):1179–1185, 2000.

[18] C. Sagüés, A. C. Murillo, J. J. Guerrero, T. Goedemé, T. Tuytelaars, and L. Van Gool. Localization with omnidirectional images using the 1d radial trifocal tensor. In *IEEE Int. Conf. on Robotics and Automation*, pages 551–556, 2006.

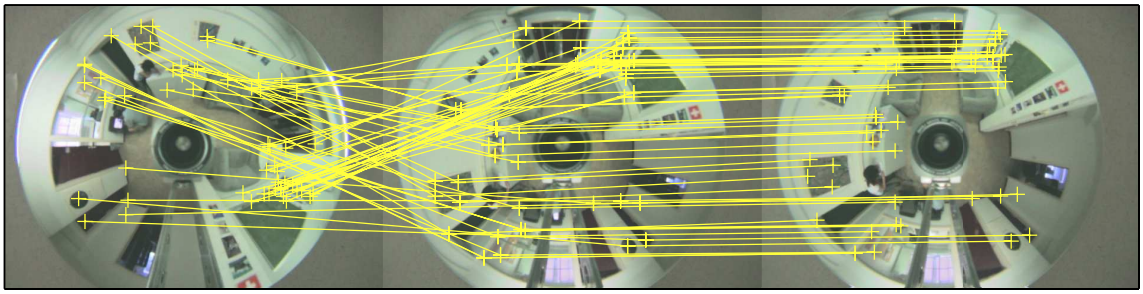
[19] K. Åström and M. Oskarsson. Solutions and ambiguities of the structure and motion problem for 1d retinal vision. *Journal of Mathematical Imaging and Vision*, 12(2):121–135, 2000.

[20] Workshop-FS2HSC-data. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2006. http://staff.science.uva.nl/ zivkovic/FS2HSC/dataset.html.

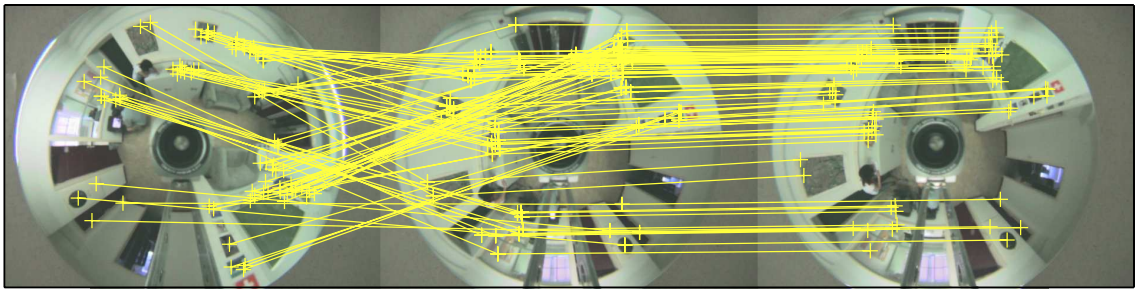
QUERY MOST SIMILAR FOUND ADJACENT IN THE VM



Radial lines. 20 matches after robust estimation (2 wrong).



SURF-64. 55 matches after robust estimation.



SIFT-128. 79 matches after robust estimation.

Fig. 5. TEST 1. Hall (room A) - images A01 A08 A09

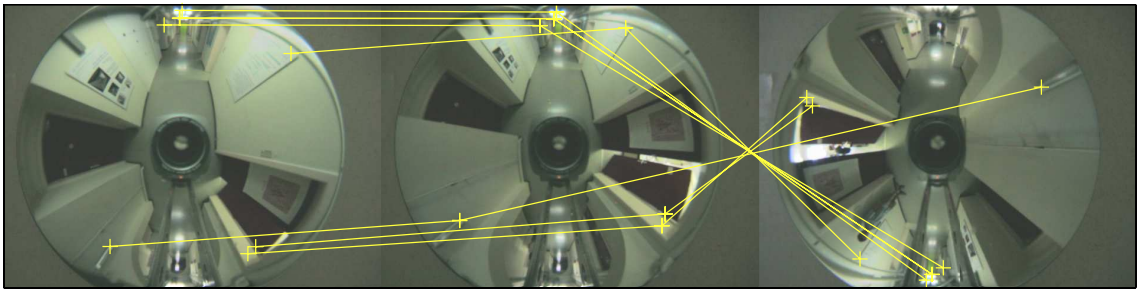
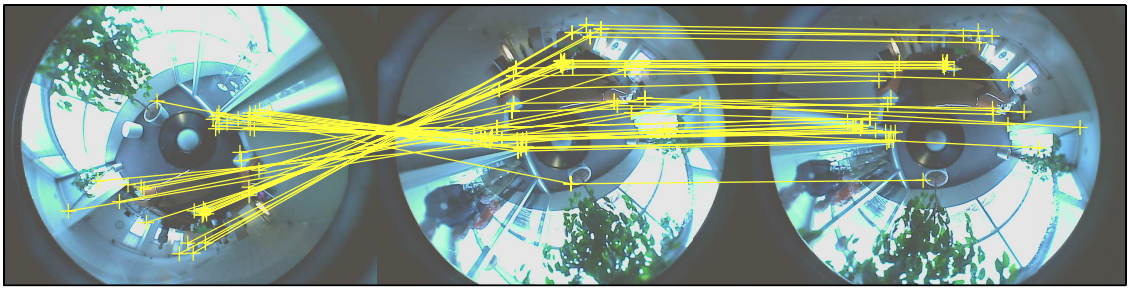


Fig. 6. TEST 2. Corridor (room D) - images D00 D02 D05: 8 robust SURF matches.



Localization	α_{21}	α_{31}	t_{21}	t_{31}
surf 64	161 ^{o}	150^o	114 ^{o}	132^o

Fig. 7. TEST 4. Frames Almere4 1125 - Almere1 500 - Almere1 550. 40 robust SURF matches.