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

Internal Report: 2007-V11

Topological and metric robot localization through computer vision techniques

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

If you want to cite this report, please use the following reference instead:
Topological and metric robot localization through computer vision techniques, A. C. Murillo, J. J. Guerrero and C. Sagüés, *Workshop.- From features to actions: unifying perspectives in computational and robot vision* held with *IEEE Int. Conference on Robotics and Automation*, pages 79-85, Rome - Italy, April 2007.

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

Topological and metric robot localization through computer vision techniques

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

***Abstract*— Vision based robotics applications have been widely studied in the last years. However, there is still a certain distance between these and the pure computer vision methods, although there are many issues of common interest in computer vision and robotics. For example, object recognition and scene recognition are closely related, which makes object recognition methods quite suitable for robot topological localization, e.g. room recognition. Another important issue in computer vision, the structure from motion problem SFM, is similar to the Simultaneous Localization and Mapping problem. This work is based on previous ones where computer vision techniques are applied for robot self-localization: a vision based method applied for room recognition and an approach to obtain metric localization from SFM algorithms for bearing only data. Several experiments are shown for both kinds of localization, room identification and metric localization, using different image features and data sets of conventional and omnidirectional cameras.**

I. Introduction

Robotic applications based on vision sensors have become widespread nowadays, but there is still a gap between these applications and the pure computer vision developments. Sometimes this separation can be due to the lack of communication between both research communities or to the divergence in their objectives. Other times this difference is due to the inadequacy of the methods for certain tasks, e.g. there are computer vision methods which can not be applied for robotic tasks due to its high computational complexity. However, this can be solved many times just with a slight adaptation of the techniques.

Many works during the last years have developed vision based methods for robotic tasks such as control [1], automatic topological map building [2], topological localization [3], or Simultaneous Localization and Mapping [4]. This work is focused on the application of computer vision techniques for robot self-localization, a fundamental issue for any autonomous device. Both topological and metric localization are taken into account, as the two of them have huge similarities with computer vision applications. On the one hand, topological localization usually consists of identifying the current location of our mobile device in a higher cognitive level than just metric units, for example identifying the room where the robot currently is. This could be also named room/scene identification. Object recognition has been an important issue in computer vision research, with many works and important results in the previous years, e.g. [5], [6] or [7], that could be adapted for scene recognition. For instance, in [8] a room identification technique was presented, that mixes range and camera information and is based on a machine learning method typically used for object classification/recognition (AdaBoost). On the other hand, the metric localization as well as the Simultaneous Localization and Mapping (SLAM) are very similar to the classical problem of Structure from Motion (SFM). The SFM algorithms provide the camera (or robot) and landmarks location from the required multi-view correspondences. Thus, they have the same goal as the SLAM. This has been studied in previous works, e.g SFM from the 1D trifocal tensor has been proved to improve bearing only SLAM initialization [9], and more recently it has been shown also the utility of SFM methods for the always difficult problem of loop closing [10], in this case using the 2D geometry for image pairs.

This paper explains a vision-based method to obtain both topological and metric localization through a hierarchical process, presented in our previous work [11]. There, global localization is obtained with respect to a visual memory (a topological map built with sorted reference images). The global localization, sometimes known as the ”kidnapped robot problem”, tries to localize the robot only with the current acquisition of the sensors, without any knowledge of previous measurements, as main difference with the continuous localization tasks. The aforementioned localization hierarchy consists of an initial less accurate localization result, in terms of topological information (room identification), which is based in the Pyramidal matching developed in [6] for object recognition. The second localization result of the hierarchy is a more accurate metric localization. It is obtained through a SFM algorithm for 1D bearing only data [12], [9] based on the 1D trifocal tensor [13]. This kind of data is intuitively extracted from images. Fig. 1 shows two examples: on the left, the orientation of point features in omnidirectional images, that is the more stable cue in that kind of images; on the right, another situation where using only 1D is convenient, the horizontal coordinate of vertical lines in conventional images, as these line segments usually have a clear orientation (x-coordinate) but they do not have too accurate tips (y-coordinate).

The outline of this paper is as follows. Next section II is divided in two parts: subsection II-A details the process

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

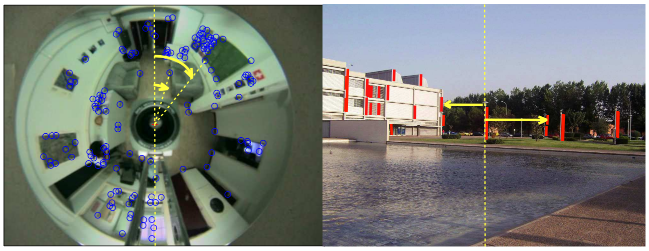


Fig. 1. Two examples of 1D bearing only data extracted from images.

used to perform the room identification and II-B explains the 1D trifocal tensor and its SFM algorithms. In section III, we can see a brief explanation of the features that will be used in our examples, and afterwards section IV shows several experiments with different kinds of images as examples of the localization results obtained with the explained techniques. Finally section V gives the conclusions of the paper.

II. Vision based hierarchical localization

This section summarizes the hierarchical localization process developed in [11], emphasizing and giving more details about the similarities between well-known computer vision tasks and some robotic ones, as well as how these computer vision methods are applied to localize the robot.

To perform both topological and metric localization in the same process has several advantages. First of all, both kinds of information are usually necessary, e.g. the topological one is more suitable to interact with users but the metric one is more accurate. The fact of dividing the process in several steps, leaving the computationally expensive ones at the end (the metric localization), helps to deal with a big amount of reference images. Fig. 2 shows a diagram of the hierarchical localization process.

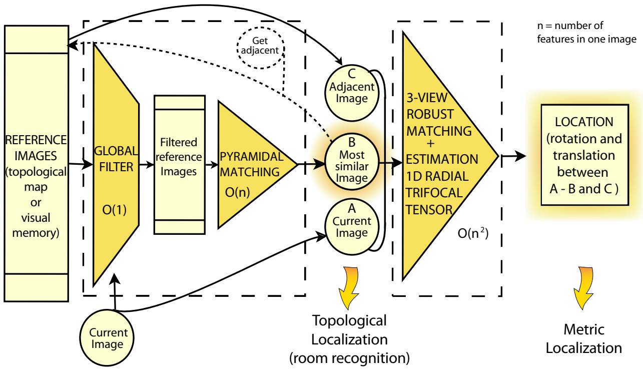

REFERENCE IMAGES (topological map or visual memory)
GLOBAL FILTER
O(1)
Filtered reference Images
PYRAMIDAL MATCHING
O(n)
Get adjacent
C Adjacent Image
B Most similar Image
A Current Image
3-VIEW ROBUST MATCHING + ESTIMATION 1D RADIAL TRIFOCAL TENSOR
O(n²)
n = number of features in one image
LOCATION (rotation and translation between A - B and C)
Current Image
Topological Localization (room recognition)
Metric Localization

Fig. 2. Diagram of the hierarchical localization process.

A. *Object Recognition* ⇒ *Room Recognition*

In the first stage, let us focus in the topological localization, that in our case consists of room identification. The goal is to localize the robot in the available topological map (or visual memory of reference images). In practice, this means to identify which room from the reference set is the most similar to the current view. In order to obtain this, a similarity evaluation algorithm is run. It is based on the matching method developed in [6], that approximates the optimal correspondences between two given sets of features and has linear computation in the number of features.

First, a pre-filtering is carried out, obtaining a global descriptor for each image, such as intensity histograms or color invariants computed all over the image pixels. Then, all reference images are compared with the current one, with regard to those global descriptors, and images with a difference over an established threshold are discarded. This step intends to reject in a fast way as many wrong candidates as possible, with a rough but quick global evaluation of the image.

After this rough initial step to discard reference images which are unprovable to match the current one, a more detailed similarity measure is obtained. Local features are extracted in the reference images that passed the pre-filtering, and the descriptor sets of all features are used to implement a *pyramid matching kernel* [6] for each image. This implementation consists of building for each image several multi-dimensional histograms (each dimension corresponds to one descriptor), where each feature occupies one of the histogram bins. The value of each feature descriptor 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 (pyramid) ψ with different levels of resolution. The similarity between two images, the current (c) and a reference one (v), is obtained by finding the intersection of the two pyramids of histograms:

$$S(\psi(c), \psi(v)) = \sum_{i=0}^{L} w_i N_i(\psi(c), \psi(v)) \ , \qquad (1)$$

with N_i the number of matches between images c and v in level i of the pyramid (features that fall in the same bin in level i of the histograms, see Fig. 3). w_i is the weight for the matches in that level, it 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 *normalized* distance obtained is

$$S_{cv} = \frac{S(\psi(c), \psi(v))}{\sqrt{S(\psi(c), \psi(c))\ S(\psi(v), \psi(v))}} \ . \qquad (2)$$

The reference image with highest similarity measure S_{cv} is chosen, it indicates the room where the robot currently is.

Notice that the matches found here are not always individual feature-to-feature matches, as the method just counts how many features fall in the same bin. The more levels we check in the pyramid the bigger are the bins, so the easier it is to get multiple coincidences in the same bin (as it can be seen in Fig. 3). Although it can be less accurate, this matching method is faster than typical matching methods based on nearest neighbour approaches, so it is very convenient for the

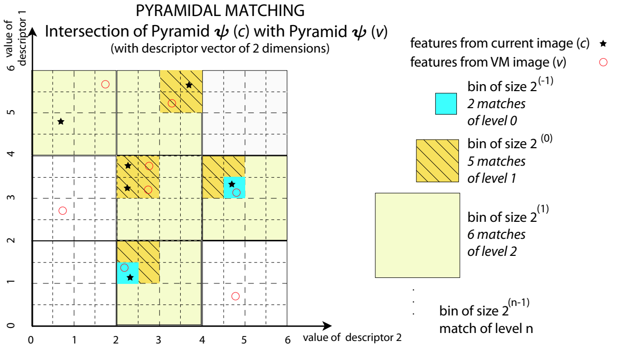


Fig. 3. Example of Pyramidal Matching, with correspondences in level 0, 1 and 2. For graphic simplification, with a descriptor of 2 dimensions.

current task that has to deal with big amounts of reference images.

B. *Structure From Motion (SFM) ⇒ Metric Localization*

As previously mentioned, the methods known in computer vision as SFM provide the simultaneous recovery of the robot and landmarks locations [14], i.e. the same goal as in SLAM. The difference could be noticed in the fact that the SLAM methods are continuous processes where the robot integrates the sensor measurements along the time, in order to obtain an accurate metric map of the environment at the end together with the robot current location with regard to that map. However, SFM algorithms are a more instantaneous procedure that gives robot and landmarks location at a certain moment. It does not use any a priori information, therefore it is very convenient for obtaining a global localization. Applications based on two view geometry have been more frequently studied in computer vision than the case of three views of 1D bearing only data, which could be convenient for robotics. This situation is the subject of this section, and it is described in Fig. 4.

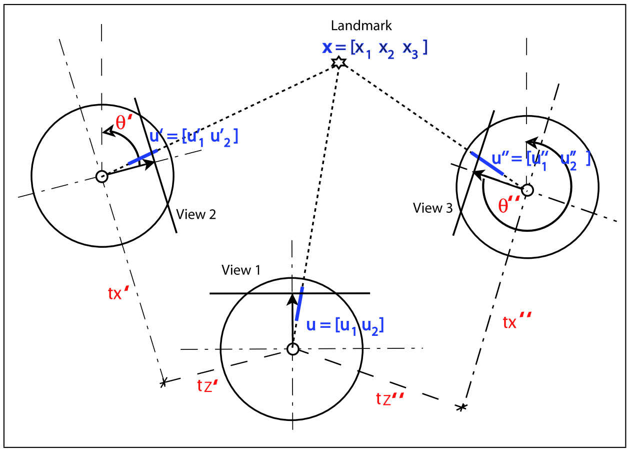


Fig. 4. Given three views of a certain scene, the goal is to obtain the relative location of the robot (θ', θ'', $\mathbf{t}' = [t'_x t'_z]$, $\mathbf{t}'' = [t''_x t''_z]$) and the position of the landmarks $\mathbf{x}$, from correspondences of bearing-only observations in three views ($\mathbf{u}$, $\mathbf{u}'$, $\mathbf{u}''$) that must be automatically matched.

To obtain the metric localization, first the 1D three view geometry constraint, the 1D trifocal tensor, has to be computed. This tensor is robustly estimated simultaneously to a robust set of three view feature correspondences, as explained in next section II-B.1. Afterwards, the robot and landmarks locations are recovered from the tensor as shown in section II-B.2.

1) Automatic Robust Matching and 1D Trifocal Tensor Computation: The 1D trifocal tensor, $\mathbf{T}$, can be computed as explained in the literature, using the trilinear constraint [13], that relates observations of a landmark in three views (u,u',u''):

$$\sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2} T_{ijk} u_i u'_j u''_k = 0. \tag{3}$$

where T_{ijk} $(i, j, k = 1, 2)$ are the eight elements of the $2 \times 2 \times 2$ 1D trifocal tensor.

The minimal number of correspondences varies in different situations. In a general case, at least seven correspondences are required, but if the two calibration constraints from [12] are included in the computations only five matches are needed. A deeper study about the tensor estimation options, and about their performance in robot applications can be found in [15] and [16].

With more matches than the minimum number required, the SVD procedure gives the least squares solution, which assumes that all the measurements can be interpreted with the same model. This is very sensitive to outliers, then robust estimation methods are necessary to avoid those outliers in the process, such as the well known *ransac* [17], which makes a search in the space of solutions obtained from subsets of minimum number of matches. This robust estimation allows to obtain simultaneously the tensor and a robust set of correspondences. It consists of the following steps:

- Extract relevant features in the three views, and perform an automatic matching process to firstly obtain a putative set of matches (*basic matching*), based on the appearance of the features in the image.
- Afterwards, the geometrical constraint imposed by the tensor is included to obtain a *robust matching* set using a *ransac* voting approach. This robust estimation efficiently rejects the outliers from the basic matching.
- Optionally, the tensor constraint can help to grow the final set of matches, obtaining new ones with weaker appearance-based similarity but fitting well the geometric constraint.

2) SFM from the 1D trifocal tensor: The camera and landmarks location parameters can be computed from the 1D trifocal tensor in a closed form. These parameters can be related to the components of the tensor by developing the elements of the projection matrixes ($\mathbf{M}, \mathbf{M}', \mathbf{M}''$). These matrixes project a 2D feature in homogeneous 2D coordinates, $\mathbf{x} = [x_1, x_2, x_3]^T$), in the $\mathcal{P}^1$ projective space, 1D images, as $\mathbf{u} = [u_1, u_2]^T$:

$$\lambda\mathbf{u} = \mathbf{M}\mathbf{x}, \qquad \lambda'\mathbf{u}' = \mathbf{M}'\mathbf{x}, \qquad \lambda''\mathbf{u}'' = \mathbf{M}''\mathbf{x}, \tag{4}$$

where λ, λ' and λ'' are scale factors.

If we suppose all the 2D features in a common reference frame placed in the first robot location, the projection matrixes relating the scene and the image features are $\mathbf{M} = [\mathbf{I}|\mathbf{0}]$, $\mathbf{M}' = [\mathbf{R}'|\mathbf{t}']$ and $\mathbf{M}'' = [\mathbf{R}''|\mathbf{t}'']$ for the first, second and third location respectively. Here, $\mathbf{R}' = \begin{bmatrix} \cos\theta' & \sin\theta' \\ -\sin\theta' & \cos\theta' \end{bmatrix}$ and $\mathbf{R}'' = \begin{bmatrix} \cos\theta'' & \sin\theta'' \\ -\sin\theta'' & \cos\theta'' \end{bmatrix}$ are the rotations and $\mathbf{t}' = [t'_x, t'_z]^T$ and $\mathbf{t}'' = [t''_x, t''_z]^T$ are the translations (Fig. 4).

We have studied two methods to recover the robot and landmarks localization from these relationships: the algorithm presented in [9], which is based on the decomposition of the tensor into two intrinsic homographies [18], and the method from [12]. Both methods give almost identical results, but the SFM algorithm from [9] is a little easier to implement (see Algorithm 1). They both provide two symmetric solutions for the location parameters, defined up to a scale for the translations. This two-fold ambiguity [12] is one of the drawbacks of using only three views to solve this problem. Once the relative location of the sensor has been estimated, the location of the landmarks can be obtained by solving the projection equations (4) for each landmark [9].

III. Local image features

Both localization processes explained in previous section are based in the analysis and matching of local image features. Choosing the feature to use is a very important practical issue, the purpose is to find the simpler and faster feature that provides us all the invariant properties required. There are many local features developed in the last years for image analysis, with the outstanding SIFT [19] as the most popular. In the literature, there are several works studying the different features and their descriptors, for instance [20] evaluates the performance of the state of the art in local descriptors, and [21] shows an study on the performance of different features for object recognition.

We have used different features for the explained algorithms in our previous works, to try to evaluate their efficiency for robotic tasks. The three kind of features used in the experiments in next section are

- Line segments, with their line support regions. We used the extraction method and descriptors explained in [11].
- SIFT. The original extraction and matching methods provided by D. Lowe [19] were used.
- SURF, a recently developed local feature, whose original extraction and matching methods [22] were used as well.

The following section shows experiments with all these features, showing some advantages and disadvantages for each one.

IV. Experiments

This section shows experimental results using the methods explained in this paper for robot localization with different image data sets. The data sets *Almere* (provided for the workshop [23]) and *data set LV* have been acquired with omnidirectional vision sensors with hyperbolic mirror. They were used and explained in more detail in [11]. The *data set ZGZ* has been acquired with a conventional camera and consists of several outdoor images in a man-made environment.

Algorithm 1 Robot Motion from the 1D Trifocal Tensor [9]

1: Decompose the trifocal tensor (computed for images 1, 2 and 3) into its intrinsic homographies. We get 6 of those homographies, but we need just three to find the epipoles, for example $\mathbf{H}^X_{32}$, $\mathbf{H}^Z_{32}$ and $\mathbf{H}^X_{12}$:

$$\mathbf{H}^X_{32} = \begin{bmatrix} -T_{112} & -T_{122} \\ T_{111} & T_{121} \end{bmatrix} \quad \mathbf{H}^Z_{32} = \begin{bmatrix} -T_{212} & -T_{222} \\ T_{211} & T_{221} \end{bmatrix}$$
$$\mathbf{H}^X_{12} = \begin{bmatrix} -T_{211} & -T_{221} \\ T_{111} & T_{121} \end{bmatrix}$$

2: Compose an homology ($\mathbf{H}$), to reproject the points of one image to the same image. The only points that will stay invariant under this reprojection are the epipoles ($e = \mathbf{H}e$), as they are the eigenvectors of $\mathbf{H}$.

$$\mathbf{H} = (\mathbf{H}^Z_{32})^{-1} * \mathbf{H}^X_{32}$$
$$[\mathbf{e}_{21} \quad \mathbf{e}_{23}] = eigenVectors(\mathbf{H})$$

with $[\mathbf{e}_{21} \quad \mathbf{e}_{23}]$ being the epipoles in the image 2 of the camera 1 and 3 respectively. A second solution will be obtained swapping both epipoles.

3: Project the epipoles in the camera 2 to the other cameras using any of the intrinsic homographies

$$\mathbf{e}_{31} = \mathbf{H}^X_{32} * \mathbf{e}_{21} \;; \quad \mathbf{e}_{32} = \mathbf{H}^X_{32} * \mathbf{e}_{23}$$
$$\mathbf{e}_{12} = \mathbf{H}^X_{12} * \mathbf{e}_{21} \;; \quad \mathbf{e}_{13} = \mathbf{H}^X_{12} * \mathbf{e}_{23}$$

4: Compute the camera motion from the epipoles as

$$\theta' = \arctan(\tfrac{e_{12}(2)}{e_{12}(1)}) - \arctan(\tfrac{e_{21}(2)}{e_{21}(1)})$$
$$[t'_x \quad t'_z] = \mathit{scale} * [e_{12}(1) \quad e_{12}(2)]^T$$

Those are the motion parameters from image 2 to 1. The parameters from image 3 to 1 (θ'', t''_x and t''_z) are computed in a similar way, by substituting in the expressions above the subindex 2 by 3.

5: Recover landmarks location from the projection equations (4) for each landmark $\mathbf{x} = (x_1, x_2, x_3)^T$:

$$\mathbf{u} \times [\mathbf{I}|\mathbf{0}]\mathbf{x} = 0$$
$$\mathbf{u}' \times [\mathbf{R}'|\mathbf{t}']\mathbf{x} = 0$$
$$\mathbf{u}'' \times [\mathbf{R}''|\mathbf{t}'']\mathbf{x} = 0$$

where × indicates the cross product. They can be explicitly developed to solve the position of the landmarks $\mathbf{x}$ defined up to an overall scale factor.

A. Room recognition

This experiment presents several results for room recognition, with respect to a reference topological map, using omnidirectional images. First, it is necessary to build the reference set, in our case named visual memory (VM). Here it was built manually, as its automatic construction was not the case of study. We used the following visual memories:
- Visual memory VM$_{LV}$: it is composed by all images from *Data set LV.*
- Visual memory VM$_1$: it is built from images from *Almere* data set - round 1.

Table I shows the results for room recognition or topological localization in several cases with the different VMs. Column $1Ok$ indicates the percentage of tests where the image found as most similar to the current, using the similarity evaluation in Sec. II-A, was correct. Since the

Pyramidal matching method is not convenient for all kind of features (specially those with very big descriptor set), a similarity evaluation using a more typical nearest neighbour based matching (NN) was performed as well.

All tests were performed with the three kind of features mentioned in Sec. III. Note that the results for SIFT shown in Table I were obtained with the NN similarity evaluation, because the ones obtained with the Pyramidal matching were worse, e.g. 60% correct classifications $(1Ok)$ for *data set LV* and much higher computational cost. This was already expected because of the big size of SIFT descriptor vector.

The time information in column T/T$_{surf}$ is just a comparative of the relative speed of the localization using each of the three evaluated features. It does not intend to evaluate their maximal speed, note that the implementations were run in Matlab and were not optimized for speed. Then the surf execution time (T$_{surf}$) is taken as reference and the others are relative to it in each case. There are three different cases studied in Table I. First, *data set LV* column includes tests using query images from this data set and the rest of the VM$_{LV}$ as reference. *Almere1$\propto$1* and *Almere4$\propto$1* columns results are from tests that used VM$_1$ for reference. The first one had query images from the same round 1, while the second one were the most difficult tests, with query images from a different round (round 4), in the same environment but with much more occlusions and noise.

The results for radial lines were definitely better with the Pyramidal matching classification, as the correctness was similar but the execution time was smaller (around 25% less for the Pyramidal matching than the NN matching). However we can observe that when the difficulty of the data set increases the performance of the radial lines decreases more than with the other features. The correct recognition rates for SURF and SIFT features were better than for lines, specially for SURF, with slightly better performance and quite lower computational times. This could be partially explained by the smaller size of the descriptor vector used here for SURF, what makes it behave better with the Pyramidal kernel construction, and also by the faster SURF extraction process.

TABLE I

ROOM RECOGNITION RESULTS.

	data set LV		*Almere1$\propto$1*		*Almere4$\propto$1*	
feature	*1 Ok*	T/T$_{surf}$	*1 Ok*	T/T$_{surf}$	*1 Ok*	T/T$_{surf}$
lines-22	90%	0.1	73%	0.2	47%	0.2
surf-36	97%	1	95%	1	67%	1
sift*-128	90%	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, the other features' results were obtained with 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 (at least 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 separate 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

Other previous works, such as [15] and [16], contain extensive experiments with simulated data to evaluate more accurately the metric localization results obtained from the 1D trifocal tensor. This section only shows experimental results with different kinds of real images.

TEST1: In this test, the 1D trifocal tensor for omnidirectional images [15] was robustly estimated using the bearing from different kinds of features correspondences. The matching results using the three previously mentioned image features are shown in Fig.5 and and the localization errors for rotation and translation direction (parameters detailed in Fig. 4) are summarized in Table II.

TABLE II

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

	rotation (^{o})		translation dir.(^{o})	
	θ'	θ''	t'	t''
lines-22	1.4	1.2	0.9	0.6
surf-64	1.2	0.9	0.9	0.4
sift-128	1.3	0.9	1	0.3

The number after each feature type shows the length of its descriptor set.

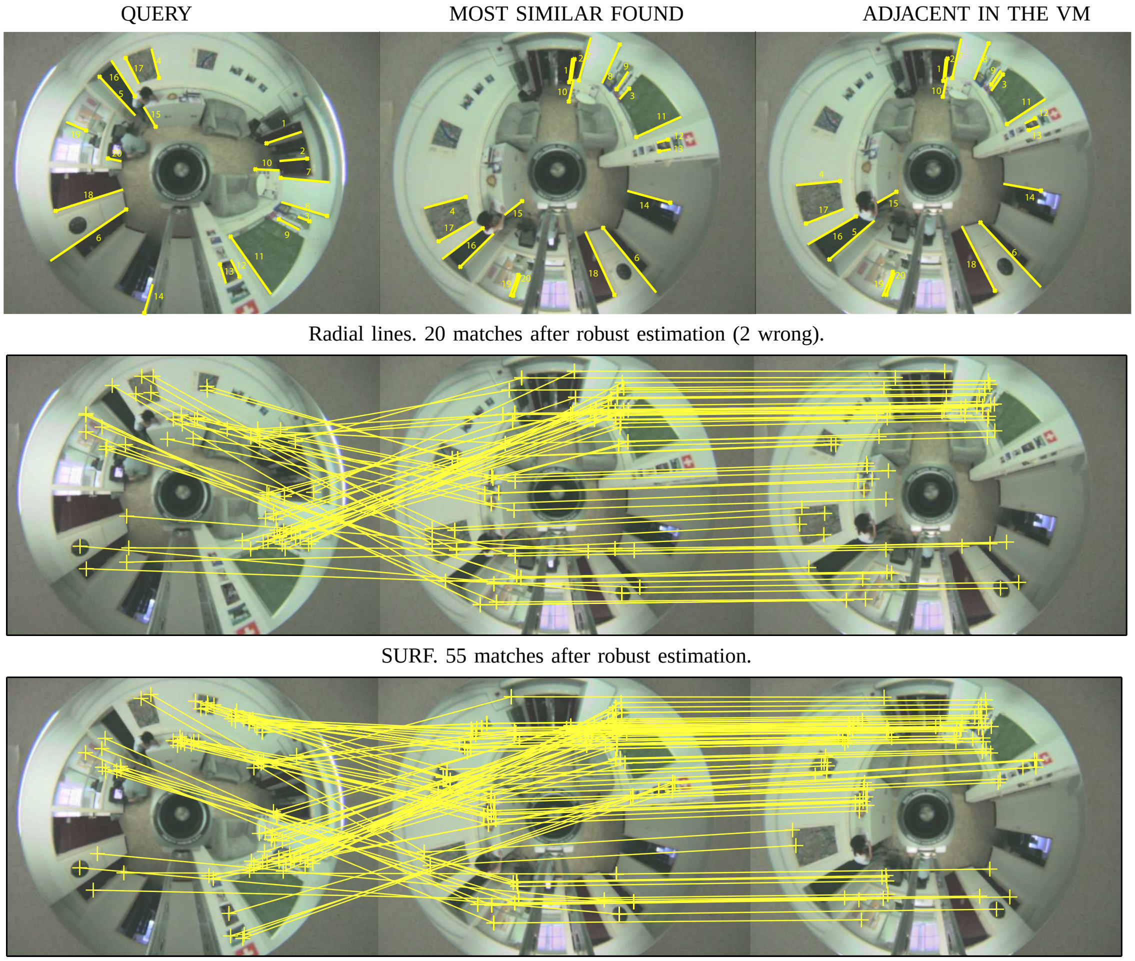
TEST 2: Fig.6 shows the results in a second test, the three view matching results and and scheme with the reconstruction of the scene landmarks. This test was similar to the previous one but this time performed with conventional images from *data set ZGZ*. Here, the 1D trifocal tensor was estimated including an extra constraint provided by a detected plane in the scene [16].

V. CONCLUSION

Some results in vision research are difficult to be used in robotic applications, probably due to the current divergence of computer vision and robotics communities. Here, we show experiments and results that tried to do accessible for robotic researchers some results in the frontier.

In the case of applying object recognition methods for scene identification, the adaptation is quite straightforward, maybe a more difficult decision is to find the most convenient kind of feature, that finds a proper balance between invariant properties and fast computations.

In the case of Structure From Motion methods applied in robot localization, most of the mathematics can be recovered from computer vision papers, and in this work we summarized its particularization to the 1D bearing-only observations with planar sensor motion, which is useful in robotics. In the research areas of omnidirectional vision systems as well as bearing-only localization and mapping, navigation or visual servoing, two view relations like the fundamental matrix or the homography have been extensively used, but the use of other multi-views constraints, like the tensors, are yet poorly studied despite its attractive properties.



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

SURF. 55 matches after robust estimation.

SIFT. 79 matches after robust estimation.

Fig. 5. TEST 1. Omnidirectional images with robust matches obtained with different features.

REFERENCES

[1] G.N. DeSouza and A. C. Kak. Vision for mobile robot navigation: A survey. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 24(2):237–267, 2002.

[2] 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.

[3] J. Košecká and F. Li. Vision based topological Markov localization. In *IEEE Int. Conf. on Robotics and Automation*, pages 1481–1486, 2004.

[4] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *the IEEE Int. Conf. on Computer Vision*, pages 1403–1410, 2003.

[5] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE Int. Conf. on Computer Vision*, pages 1150–1157, 1999.

[6] 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.

[7] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):416–431, 2006.

[8] O. Martínez Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard. Supervised semantic labeling of places using information extracted from sensor data. *Robotics and Autonomous Systems*, 2007. In press.

[9] F. Dellaert and A. Stroupe. Linear 2d localization and mapping for single and multiple robots. In *IEEE Int. Conf. on Robotics and Automation*, pages 688–694, 2002.

[10] P. Newman, D. Cole, and Kin Ho. Outdoor slam using visual appearance and laser ranging. In *IEEE Int. Conf. on Robotics and Automation*, pages 1180–1187, 2006.

[11] 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.

[12] 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.

[13] 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.

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

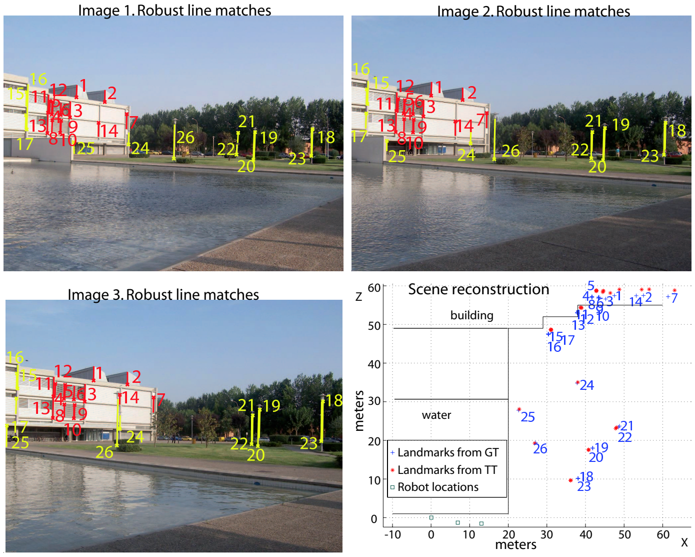
[15] C. Sagüés, A. C. Murillo, J. J. Guerrero, T. Goedemé, T. Tuytelaars, and L. Van Gool. Localization with omnidirectional images using the 1d radial trifocal tensor. In *IEEE Int. Conf. on Robotics and Automation*, pages 551–556, 2006.

[16] A. C. Murillo, J. J. Guerrero, and C. Sagüés. Robot and landmark localization using scene planes and the 1d trifocal tensor. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2070–2075, 2006.

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

[18] A. Shashua and M. Werman. Trilinearity of three perspective views and its associate tensor. In *IEEE Int. Conf. on Computer Vision*, pages 920–925, 1995.

[19] D. G. Lowe. Distinctive image features from scale-invariant key-



	Robot localization error				*Landmarks reconstruction error*	
	rotation		*transl. dir.*		*mean*	*mean*
	θ'	θ''	t'	t''	*2D x-coord.*	*2D z-coord.*
lines	0.17^o	0.5^o	2.9^o	3.4^o	0.5 m.	1.5 m.

Fig. 6. TEST 2. Top: Outdoor images with line robust matches (coplanar lines [1..14] marked in red) and scene reconstruction scheme with robot locations and landmarks locations obtained through the trifocal tensor (in red *) and from the ground truth motion (in blue +). Bottom: robot and landmarks localization errors.

points. *Int. Journal of Computer Vision*, 60(2):91–110, 2004, http://www.cs.ubc.ca/ lowe/keypoints/.

[20] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[21] K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *the IEEE Int. Conf. on Computer Vision*, pages 1792–1799, 2005.

[22] 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/.

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