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

Internal Report: 2008-V12

Topological and metric robot localization through computer vision techniques[1]

A.C. Murillo, J.J. Guerrero, 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, C. Sagüés *Unifying Perspectives in Computational and Robot Vision, Cap. 8, Lecture Notes Electrical Engineering*, Vol. 8, pages 113-127, 2008).

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

Chapter 1
Topological and metric robot localization through computer vision techniques

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

1.1 Introduction

Nowadays, robotic applications based on vision sensors have become widespread, 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 their 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 [5], automatic topological map building [23], topological localization [10], or Simultaneous Localization and Mapping [3]. This work is focused on the application of computer vision techniques for robot global 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 also be named room/scene identification. Object recognition is an important issue in computer vision research, with many works and important results in the previous years, e.g. [11], [8] or [19], that could be adapted for scene recognition. For instance, in [12] a room identification technique was presented, that mixes range and camera information and is based on a 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)

A. C. Murillo · J. J. Guerrero · C. Sagüés
University of Zaragoza, I3A - Department of Informatics and Systems Engineering, 50010 Zaragoza, Spain, e-mail: acm@unizar.es

are very similar to the classical computer vision problem of Structure from Motion (SFM). The SFM algorithms provide the camera (or robot) and landmarks location from the required minimum number of 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 [4], and more recently it has been shown also the utility of SFM methods for the always difficult problem of loop closing [18], 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 [17]. 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", intends to localize the robot only with the current acquisition of the sensors, without any knowledge of previous measurements, oppositely to the continuous localization tasks. The aforementioned localization hierarchy consists of an initial less accurate localization result, in terms of topological information (room identification), which applies object recognition techniques. 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 [1], [4] based on the 1D trifocal tensor [6]. This kind of data is intuitively extracted from images, Fig. 1.1 shows two examples of 1D bearing only data. 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 accurate tips (y-coordinate).

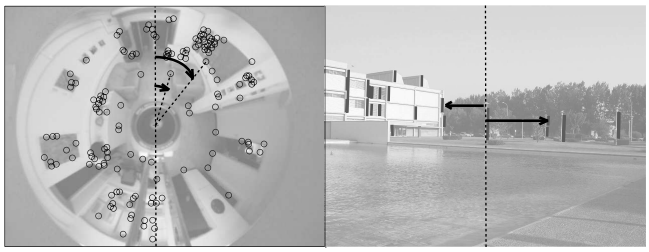

Fig. 1.1 Two examples of 1D bearing only data extracted from images.

The outline of this paper is as follows. Next section 1.2 is divided in two parts: subsection 1.2.1 details the process used to perform the room/scene recognition and subsection 1.2.2 explains the 1D trifocal tensor estimation and its SFM algorithms. In section 1.3, a brief description of the features that have been studied in our examples is given, followed by section 1.4 with several examples of localization results obtained applying the explained techniques. Finally section 1.5 concludes the work.

1.2 Vision based hierarchical localization

This section summarizes the hierarchical localization process developed in [17], including some small improvements in the process and emphasizing the similarities between well-known computer vision tasks and some robotic ones, as well as how these computer vision methods are applied to robot localization.

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 designing a hierarchical process, leaving the computationally expensive steps at the end (those needed for a metric localization), helps to deal efficiently with a big amount of reference images. The diagram in Fig. 1.2 summarizes the hierarchical localization process, whose two main stages are detailed in next subsections.

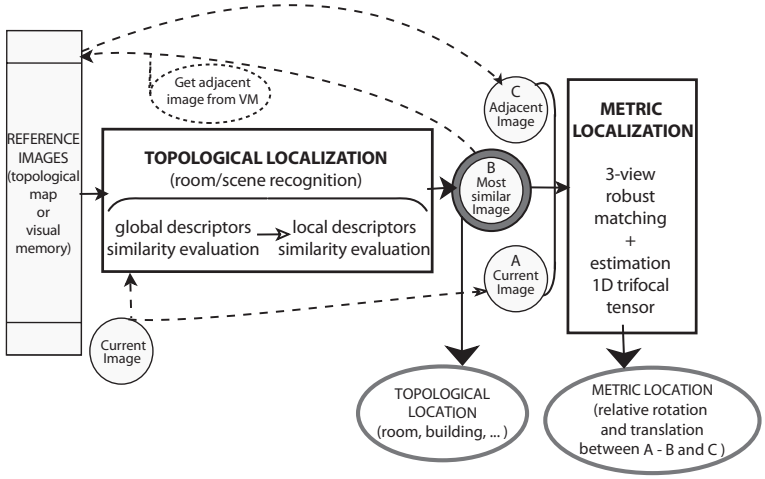

Fig. 1.2 Diagram of the hierarchical localization process.

1.2.1 Object Recognition ⇒ Room/Scene Recognition

Firstly, let us focus in the topological localization, which corresponds to the first stage of the hierarchical process. In our case of study, this topological localization consists of room identification indoors and of building recognition outdoors. 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 image from the reference set is the most similar to the current view, which will point the topologi-

cal location. In order to obtain this, a similarity evaluation algorithm is run in two stages: a first one using simple global image descriptors, and a second one using local image features.

1.2.1.1 Global Image Descriptors Evaluation

First, a pre-filtering is carried out evaluating a global descriptor, such as color histograms or color invariant moments, which is computed for each image over all its pixels. Then, all reference images are compared with the current view with regard to the chosen global descriptor, and a probability to be the current room/location is estimated for each reference image. Images with lower probability than the 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 appearance of the images.

1.2.1.2 Local Image Descriptors Evaluation

After the rough initial step to discard reference images which are unprovable to match the current one, a more detailed similarity measure is obtained evaluating local image features descriptors. Two approaches were studied in our previous work [17] for this task, a typical nearest neighbour (*NN*) based matching and the pyramidal matching method developed in [8], that approximates the optimal correspondences between two given feature sets in linear time with the number of features.

NN-based approach. The first approach for image similarity evaluation is based on a local feature nearest neighbour matching. This process has quadratic (n^2) or $n\ log(n)$ computational cost with the number of features (n), depending on the implementation chosen. We use a typical approximate nearest neighbour implementation that has the lower complexity. After the matching, a probability P_v is estimated for each reference image (v) processed at this stage, which depends on the similarity S_{cv} between the reference image and the current view (c):

$$P_v = e^{\frac{-(1-S_{cv})}{\sigma_s}}, \tag{1.1}$$

being σ_s the variance among the similarity values between all reference images and the current one. In this approach, the similarity measure S_{cv} is obtained from the distance between the two views d_{cv}: $S_{cv} = \frac{1}{d_{cv}+1}$, being $d_{cv} = md_e + Fz$, with m the number of matches, d_e the average Euclidean distance between each pair of matched features, F the number of non matched features and z a penalty for it.

Pyramid-based approach. In the second image similarity evaluation approach based on local features studied, the descriptor sets of all features are used to implement a *pyramid matching kernel* [8]. 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 corresponding histogram resolution, which gives the coordinates of the bin corresponding to that feature. Several levels of histograms are defined, and 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. Once these pyramids are built, the similarity S between two images, the current one (c) and a reference image (v), is obtained from the intersection of the two pyramids of histograms as explained in next eq. (1.2). This operation is quite efficient, with linear complexity in the number of features:

$$S(\psi(c),\psi(v)) = \sum_{i=0}^{L} w_i N_i(\psi(c),\psi(v)) \,, \tag{1.2}$$

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. 1.3). w_i is the weight for the matches in level i and is the inverse of the current bin size (2^i). L is the level where all features fall in the same bin, e.g., Fig. 1.3 example has $L = 3$. This similarity measure is divided by a factor determined by the self-similarity score of each image, in order to avoid giving advantage to images with bigger feature sets, so the *normalized* similarity measure obtained with this approach is

$$S_{cv} = \frac{S(\psi(c),\psi(v))}{\sqrt{S(\psi(c),\psi(c))\, S(\psi(v),\psi(v))}} \,. \tag{1.3}$$

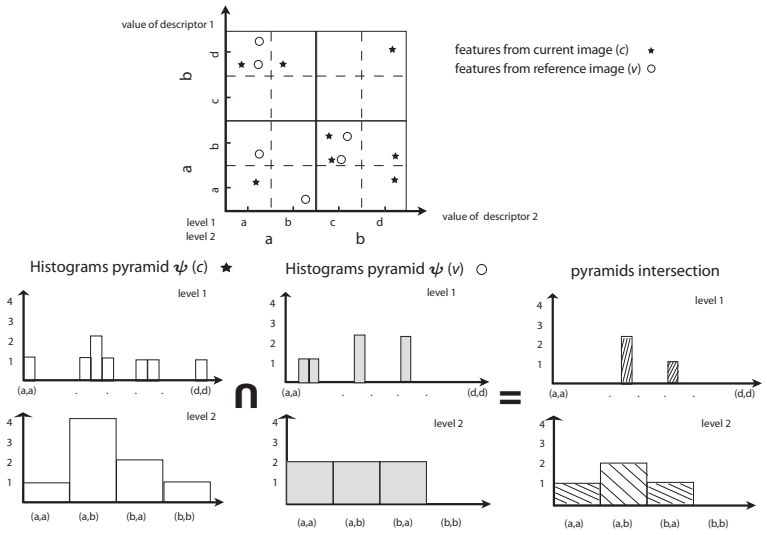

Fig. 1.3 Local Features Pyramids construction and matching (intersection). Top: plot with the features from two images (c and v). Bottom: histograms pyramids (ψ) of both images and their intersection (for graphic simplification, only two levels evaluation and feature descriptor of 2 dimensions).

Similarly to the previous studied approach, a probability for each reference image being the current location is estimated based on this S_{cv} (see eq. 1.1).

Notice that the matches found with this approach 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 the bins are, so the easier it is to get multiple coincidences in the same bin (as it can be seen in Fig. 1.3). Although this matching method can be less accurate, it can also be faster than typical matching methods based on nearest neighbour approaches, so it is very convenient for the current task when it is necessary to deal with big amounts of reference images.

Finally, to obtain the topological localization result, using either similarity evaluation approach shown in this section, we select the reference location (image) with higher probability (P_v) as the current topological location. If our process would finish with this topological localization, it would be convenient to make a robust selection from the x most probable locations, for example imposing a multi-view geometry constraint [9] to theirs sets of correspondences or making a voting process with these images. However, if we carry on the whole hierarchical localization process, the method continues with a last step based pursuing a more accurate metric localization, which is based on a robust estimation of geometry constraints that will be explained in next subsection. Then, if this following robust estimation fails, the process could go back and pick up the reference image with second higher probability.

1.2.2 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 from feature correspondences in multiple views [9], i.e., similar goals as in the SLAM problem. 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, or recovering a lost robot. Applications based on two view geometry have been more frequently studied in computer vision than the case of three views, which could be convenient for robotics for example in the case of using 1D bearing only data. This situation is the subject of this section and is described in Fig. 1.4.

To obtain the metric localization in the case of study, the 1D three view geometry constraint (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 1.2.2.1. Afterwards, the robot and landmarks locations are recovered from the tensor as shown in section 1.2.2.2.

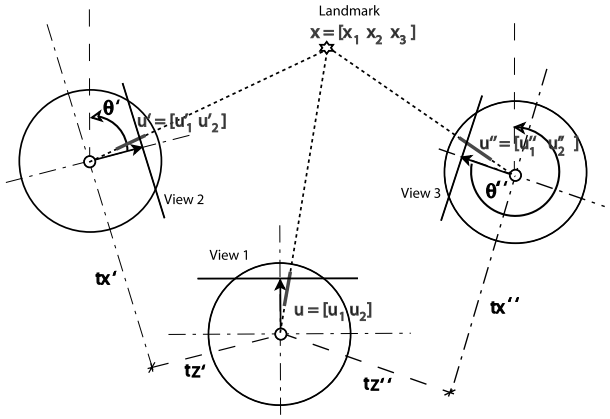

Fig. 1.4 SFM with three-view 1D geometry (the 1D trifocal tensor): with at least five correspondences of bearing-only observations in three views ($\mathbf{u}$, $\mathbf{u}'$, $\mathbf{u}''$), we can estimate the 1D tensor and extract from it the relative location of the robot (θ', θ'', $\mathbf{t}' = [t'_x t'_z]$, $\mathbf{t}'' = [t''_x t''_z]$) and the position of the landmarks $\mathbf{x}$.

1.2.2.1 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 [6], that relates observations of a landmark in three views (u,u',u''):

$$\sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2} T_{ijk} u_i u'_j u''_k = 0. \tag{1.4}$$

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 [1] 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 [20] and [15].

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 [7], 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.

1.2.2.2 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 $\mathscr{P}^1$ projective space, 1D images, as $\mathbf{u}=[u_1,u_2]^T$:

$$\lambda\mathbf{u}=\mathbf{M}\mathbf{x}, \qquad \lambda'\mathbf{u}'=\mathbf{M}'\mathbf{x}, \qquad \lambda''\mathbf{u}''=\mathbf{M}''\mathbf{x}, \tag{1.5}$$

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. 1.4).

We have studied two methods to recover the robot and landmarks locations from these relations: the algorithm presented in [4], which is based on the decomposition of the tensor into two intrinsic homographies [21], and the method from [1]. Both methods give almost identical results, but the SFM algorithm from [4] 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 [1] is one of the drawbacks of using only three views to solve this problem, it can be solved using a fourth view or some additional information such as odometry. Once the relative location of the sensor has been estimated, the location of the landmarks can be obtained by solving the projection equations (1.5) for each landmark [4].

Algorithm 1 (Robot Motion from the 1D Trifocal Tensor [4]).

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 $\mathbf{H}^X_{12}$:

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

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 $\mathbf{H}$.

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

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 image 2 to the other cameras using any of the intrinsic homographies

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

4. Compute the camera motion from the epipoles as

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

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, substituting in the expressions above the subindex 2 by 3.

5. Recover landmarks location from the projection equations (1.5) 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 $\times$ 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.

1.3 Local image features

The main parts of the localization processes explained in previous section are based on 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 simplest and fastest feature that provides all the invariant properties required. There are many local features developed in the last years for image analysis, with the outstanding SIFT [11] as the most popular. In the literature, there are several works studying the different features and their descriptors, for instance [14] evaluates the performance of the state of the art in local descriptors, and [13] shows an study on the performance of different features for object recognition.

We have used different features with the developed algorithms in our previous works, to try to evaluate their efficiency for the aimed 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 [17].
- SIFT (Scale Invariant Feature Transform). The original code provided by D. Lowe [11] was used.
- SURF (Speeded Up Robust Features), a recently developed local feature, whose original extraction method provided by the authors [2] was used as well, which allows a flexible descriptor length. We will use the simpler descriptor, SURF-36,

(36 descriptors per feature) for the topological localization, since at that stage speed is more important. However for the metric localization, where higher accuracy is necessary, we will extract the 64-descriptors features.

The following section shows localization experiments using all these features, showing some advantages and disadvantages of using one or another.

1.4 Experiments

This section shows experimental results using the methods explained in this work for robot localization with different image data sets. The data sets used are *Almere* (publicly available [22]) and *data set LV*, which were acquired with omnidirectional vision sensors with hyperbolic mirror and were explained in more detail in [17], and another data set of conventional images, the *data set ZGZ*, that consists of 630 outdoor images from an urban environment.

1.4.1 Topological Localization: Room/Building Recognition

This section presents several results applying the explained methods for room recognition in case of indoor omnidirectional images, and for building/scene recognition in case of outdoor conventional images.

Room recognition with omnidirectional images. In a first topological localization experiment, robot localization is performed with respect to a reference topological map using omnidirectional images. Initially it is necessary to build the reference map, also named visual memory (VM). Here it was built manually, grouping the images in rooms, as its automatic construction was not the case of study. We used both data sets of omnidirectional images mentioned previously (*Almere* and *data set LV*), that where divided in reference images and test images. In case of the first data set, images were frames from robot tour videos, then every 5th even image was used as reference, and every 5th odd image was used as test. From the second data set, as images were already sparser, every image was localized with regard to the rest.

In this experiment, the global descriptor used is based on color invariant moments, similar to those used in [17]. A summary of the room recognition rates obtained for the different local features studied (Sec. 1.3) is shown in Table 1.1. Those results were studied in more detail in [16]. The time information in column Time/Time$_{surf}$ in this experiment 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 experiments were run in Matlab and were not optimized for speed. Then, the surf execution time (Time$_{surf}$) is taken as reference and the others are relative to it. Column % *Ok* shows the percentage of tests where the image selected as most similar to the current one was correct, and the second col-

umn shows the matching approach used in that case. Both of them (NN-based and Pyramidal-based) performed similarly for lines and SURF. However, using SIFT much better results were obtained with the NN-based approach (only a 60% of correct classifications with the Pyramidal based approach while the 85% was obtained with the NN-based one). This result is not surprising, since the Pyramidal matching method is not convenient for features with very long descriptor sets.

Table 1.1 Room recognition results (the number after each feature type shows the length of its descriptor set).

feature used	matching approach	% Ok	Time/Time$_{surf}$
lines-22	Pyramidal-based	81%	0.1
surf-36	Pyramidal-based	96%	1
sift-128	NN-based	85%	3

These results are the average results using test images, from both *LV* and *Almere* data sets, that were obtained under similar conditions (illumination, noise, occlusions, ...) than the reference images and they do not include big baselines between images, therefore the performance is acceptable with the three studied features. However, when the test images have higher variances, as shown in some of the experiments in [16], radial lines performance decreases dramatically. As it was also concluded there, the best compromise between correctness in the results and efficiency in the localization process was obtained with the SURF features.

This topological localization approach is quite robust, as we reduced the size of the reference images to the half and the performance stayed similar to the presented results. So reducing the reference image set is not the main problem for the correctness in the topological localization, but it can be a problem for the accuracy in a next step towards the metric localization, because a big variation between reference and current images can makes the robust local feature matching fail, as it will be seen in next section 1.4.2 results.

Building recognition with conventional images. In a second topological localization experiment, we will use only SURF features, as they showed the best performance versus execution time trade-off. The data set used in this case consists of outdoors conventional images from a urban environment. Now the localization consists of recognizing the scene (building) where the current view is taken, as part of an autonomous urban guide. The visual memory contains 600 reference images obtained with a conventional camera, from which 100 images correspond to labelled buildings (10 buildings, both day and night images). The test set consist of other 30 images from the labelled buildings, obtained from different days.

In this experiment the global descriptor used was a histogram computed over the Hue color band of the images (in the HSV image color space). Table 1.2 shows the average correct recognition rates (*%Ok*) and execution times for all tests. Most failures were due to the pre-filtering, i.e., the global descriptor evaluation discarded too many or all images from the correct reference location. Then, skipping the global

descriptor based pre-filtering increases the recognition rates close to 100%. However, as the reference set is quite big, it is necessary to include the pre-filtering step if we require an efficient answer. Note that although the matching time (column *matching time*) using the NN-based approach is higher than using the Pyramidal one, it requires being able to pre-load the pyramidal matching structures, otherwise the whole process execution time ($matching + structures\ build$) is much lower for the NN-based method. Then, the Pyramidal matching is convenient only if it is possible to load the Pyramid search structures in advance, otherwise the advantage of the linear complexity matching method is hidden by the high cost of building its data structures. In this experiment, we consider a urban tourist guide whose goal was to recognize the building in the current view, therefore we can stop at this point (topological localization) of the hierarchical process. As mentioned before, it is convenient to apply a geometric constraint to the reference images with higher probability of being the current location. Here we perform a RANSAC based estimation of a Fundamental matrix and robust set of matches [9] between the current view and the five reference images with higher P_v. Then, the P_v is re-computed only with the correspondences that passed this robust estimation [9] and based on this we choose the current location. Without this robust selection process the rate of correct recognitions shown in Table 1.2 decreased from 90% to 80%. Two examples of the reference image selected as most probable location with the two different approaches are shown in Fig. 1.5.

Table 1.2 Building recognition results using SURF features with a 600 reference images set. Both approaches using Hue histograms as global descriptor and two view geometric constraints to make a robust selection of the current location.

similarity evaluation approach	*%Ok*	*matching time*	*matching + structures build time*
NN-based	90 %	1.35 s.	
Pyramidal-based	90 %	0.2 s.	12.45 s.

1.4.2 Metric Localization

Other previous works, such as [20] and [15], contain extensive experiments with simulated data to evaluate more accurately the metric localization results obtained from the 1D trifocal tensor. This section shows an example of metric localization to remark some of the conclusions previously mentioned. In this experiment, the 1D trifocal tensor for omnidirectional images [20] was robustly estimated using the bearing from local features correspondences. An example of the robust matching obtained with radial lines or with SURF is shown in Fig. 1.6, together with the 2D reconstruction of the scene (the matched features and the robot locations) obtained from the tensor estimated in each case. Results using SIFT, both for matching and reconstruction, were very similar to SURF's ones.

Fig. 1.5 Two examples of building recognition. Top: an example where both approaches (NN-based and Pyramidal-based) succeed; Bottom: example with night images, where the Pyramidal-based approach failed.

A more detailed evaluation of the same experiment is summarized in Table 1.3, with the localization errors for rotation, translation direction (parameters detailed in Fig. 1.4) and landmarks location. Any of the three local feature studied (lines, SURF and SIFT) provides accuracy enough for the metric localization, as long as the variance between the used images does not make its matching process fail. The less robust features are the radial lines, as they are not able to deal with as big image changes as SIFT or SURF. In this example we can observe the fact that radial lines provide a less stable matching. In some executions they give a good result (see results in Fig. 1.6), but other times its robust estimation process fails because there were too many outliers in the initial matching sets. Then, the process incorrectly includes some wrong correspondence in the final trifocal tensor estimation, what makes the accuracy of the corresponding metric localization decrease.

Table 1.3 Robot metric localization errors estimating the 1D tensor with different features (statistics from 50 executions). The number after each feature type shows the length of its descriptor vector.

Feature used	*Robot localization error*				*Landmarks reconstruction error (m)*	
	rotation		*transl. direction*		*mean*	*mean*
	θ' (std)	θ'' (std)	t' (std)	t'' (std)	in x-coord. (std)	in z-coord. (std)
lines-22	0.9^o (6)	1.8^o (11)	7.5^o (25)	6.5^o (13)	0.7 (1)	1.6 (2.8)
surf-64	1.1^o (0.1)	2.1^o (0.1)	0.4^o (0.4)	8.7^o (0.3)	0.2 (0.02)	0.2 (0.06)
sift-128	0.8^o (0.1)	1.9^o (0.1)	0.9^o (0.5)	8.7^o (0.3)	0.1 (0.01)	0.2 (0.01)

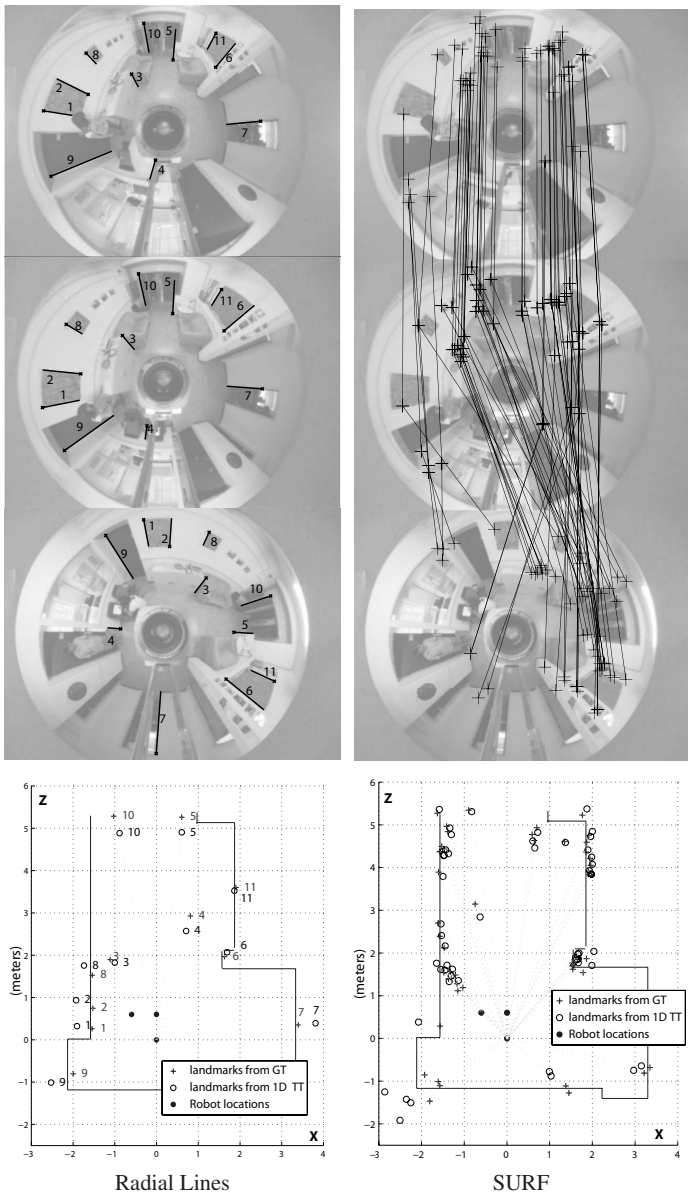

Fig. 1.6 Omnidirectional images (from *dataSet LV*) with robust matches and reconstruction of the scene from the motion parameters obtained with 1D tensor estimated with those matches (landmarks from 1D TT) or with the ground truth motion (landmarks from GT).

1.5 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 intend 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.

Acknowledgements Thanks to Oscar Calderón for his contribution with the building recognition experiments. This work was supported by projects DPI2003-07986, DPI2006-07928 and IST-1-045062-URUS-STP.

References

1. Åström, K., Oskarsson, M.: Solutions and ambiguities of the structure and motion problem for 1d retinal vision. Journal of Mathematical Imaging and Vision **12**(2), 121–135 (2000)
2. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European Conference on Computer Vision, pp. 404–417 (2006). Http://www.vision.ee.ethz.ch/ surf/
3. Davison, A.J.: Real-time simultaneous localisation and mapping with a single camera. In: the IEEE Int. Conf. on Computer Vision, pp. 1403–1410 (2003)
4. Dellaert, F., Stroupe, A.: Linear 2d localization and mapping for single and multiple robots. In: IEEE Int. Conf. on Robotics and Automation, pp. 688–694 (2002)
5. DeSouza, G., Kak, A.C.: Vision for mobile robot navigation: A survey. IEEE Trans. on Patt. Analysis and Machine Intelligence **24**(2), 237–267 (2002)
6. Faugeras, O., Quan, L., Sturm, P.: 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)
7. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Comm. of the ACM **24**, 381–395 (1981)
8. Grauman, K., Darrell, T.: The pyramid match kernels: Discriminative classification with sets of image features. In: IEEE Int. Conf. on Computer Vision, pp. 1458–1465 (2005)
9. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
10. Košecká, J., Li, F.: Vision based topological Markov localization. In: IEEE Int. Conf. on Robotics and Automation, pp. 1481–1486 (2004)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. Journal of Computer Vision **60**(2), 91–110 (2004). Http://www.cs.ubc.ca/ lowe/keypoints/
12. Martínez Mozos, O., Triebel, R., Jensfelt, P., Rottmann, A., Burgard, W.: Supervised semantic labeling of places using information extracted from sensor data. Robotics and Autonomous Systems **5**(55), 391–402

13. Mikolajczyk, K., Leibe, B., Schiele, B.: Local features for object class recognition. In: the IEEE Int. Conf. on Computer Vision, pp. 1792–1799 (2005)
14. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis & Machine Intelligence **27**(10), 1615–1630 (2005)
15. Murillo, A.C., Guerrero, J.J., Sagüés, C.: Robot and landmark localization using scene planes and the 1d trifocal tensor. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2070–2075 (2006)
16. Murillo, A.C., Guerrero, J.J., Sagüés, C.: Surf features for efficient robot localization with omnidirectional images. In: IEEE/RSJ Int. Conf. on Robotics and Automation, pp. 3901–3907 (2007)
17. Murillo, A.C., Sagüés, C., Guerrero, J.J., Goedemé, T., Tuytelaars, T., Van Gool, L.: From omnidirectional images to hierarchical localization. Robotics and Autonomous Systems **55**(5), 372–382 (2007)
18. Newman, P., Cole, D., Ho, K.: Outdoor slam using visual appearance and laser ranging. In: IEEE Int. Conf. on Robotics and Automation, pp. 1180–1187 (2006)
19. Opelt, A., Pinz, A., Fussenegger, M., Auer, P.: Generic object recognition with boosting. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(3), 416–431 (2006)
20. Sagüés, C., Murillo, A.C., Guerrero, J.J., Goedemé, T., Tuytelaars, T., Van Gool, L.: Localization with omnidirectional images using the 1d radial trifocal tensor. In: IEEE Int. Conf. on Robotics and Automation, pp. 551–556. Orlando, USA (2006)
21. Shashua, A., Werman, M.: Trilinearity of three perspective views and its associate tensor. In: IEEE Int. Conf. on Computer Vision, pp. 920–925 (1995)
22. Workshop-FS2HSC-data: IEEE/RSJ International Conference on Intelligent Robots and Systems (2006). Http://staff.science.uva.nl/ zivkovic/FS2HSC/dataset.html
23. Zivkovic, Z., Bakker, B., Kröse, B.: Hierarchical map building using visual landmarks and geometric constraints. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 7–12 (2005)