

From Omnidirectional Images to Hierarchical Localization [★]

A.C. Murillo ^{a,*} C. Sagiúes ^a J.J. Guerrero ^a T. Goedemé ^b
T. Tuytelaars ^b L. Van Gool ^b

^a*DIIS - I3A, University of Zaragoza, Spain*

^b*PSI-VISICS, University of Leuven, Belgium*

Abstract

We propose a new vision-based method for global robot localization using an omnidirectional camera. Topological and metric localization information are combined in an efficient, hierarchical process, with each step more complex and accurate than the previous one but evaluating less images. This allows to work with large reference image sets in a reasonable amount of time. Simultaneously, thanks to the use of 1D three view geometry, accurate metric localization can be achieved based on just a small number of nearby reference images. Thanks to the wide baseline features used, the method deals well with illumination changes and occlusions, while keeping the computational load small. The simplicity of the radial line features used speeds up the process without affecting the accuracy too much. We show experiments with two omnidirectional image data sets to evaluate the performance of the method and compare the results using the proposed radial lines with results from state-of-the-art wide-baseline matching techniques.

Key words: topological + metric localization, hierarchical methods, radial line matching, omnidirectional images.

1 Introduction

A fundamental issue for an autonomous device is self-localization. In general, we can distinguish three localization tasks: 1) global localization (also known

[★] This work was supported by the projects DPI2003-07986, DPI2006-07928, IST-1-045062-URUS-STP, IUAP-AMS, IWT and FWO-Vlaanderen.

* Corresponding author.

Email address: acm@unizar.es (A.C. Murillo).

as the kidnapped robot problem), 2) continuous localization, and 3) simultaneous localization and map building (SLAM). The work presented in this paper is focused on the initial or global localization, where the robot has to self-localize without any historical information, i.e. it only disposes of the data currently captured by the device and some kind of (precomputed) reference map of the environment. The solution we propose is a purely vision-based one, and uses the simplest reference map possible in that case, namely a set of more or less organized images.

Several recent publications have focused on this problem of global localization using visual reference information, e.g. [1] and [2]. They point out the importance of this initial localization. Sometimes, global localization is a goal in itself. For instance, when a museum-guide is started by a visitor somewhere in a museum to receive information about that particular location, the initial global localization is all that counts. Sometimes, global localization information is used as an initialization step for continuous localization or simultaneous localization and map building methods. Recently, global localization results have also been used for submap alignment [3].

The representation of the "world" information in the reference set (or map) must allow the robot to localize itself with as much accuracy as needed for the task to be performed. For instance, for navigation the robot needs to localize itself accurately, and therefore to correct the trajectory due to odometry errors. In order to achieve this, metric localization information is needed. In other situations, the robot just needs topological localization information, less precise but indeed more intuitive information to communicate with humans, e.g. identifying in which room it is.

We propose a hierarchical vision based method that allows both topological and metric global localization, with respect to a set of reference omnidirectional images, which we coin our Visual Memory (VM). This VM consists of a database of sorted omnidirectional reference images, including some topological information (the room where they belong) as well as some metric information (relative positions between neighboring views stored). This kind of topological maps can be built from sub-sampling a robot video stream captured in exploration, as done in our experiments or in [4], or with a more complex processing of the video stream, e.g. in order to deal with similar locations in the environment [5] or to build automatically a minimal topological map [6]. The two first steps of our hierarchical process provide the topological localization (the current room identification). In order to find the most similar room in the VM to the current view, a similarity evaluation algorithm is run. It is based on the Pyramidal matching kernels introduced in [7]. They showed very good results in object classification, and here we show its nice performance also for scene identification. In the final and third step, the metric localization relative to the VM is reached. There we use local feature correspondences in

three views to simultaneously estimate a robust set of matches and the 1D trifocal tensor. The relative location between views can be obtained with an algorithm based on the 1D trifocal tensor for omnidirectional images [8]. The 1D trifocal tensor was previously studied in [9] and [10].

Both hierarchical methods and omnidirectional vision are subjects of interest in the robotics field. Omnidirectional vision has become widespread over the last few years, and has many well-known advantages, but unfortunately also some extra difficulties compared to conventional images. There is much recent work in this field, using all kind of omnidirectional images, e.g. images from conic mirrors applied in [11] for map-based navigation, or localization based on panoramic cylindrical images composed of mosaics of conventional ones [12]. Hierarchical approaches are also of great interest, especially to deal with big reference sets in an efficient way. The hierarchy consists of several steps, of which the initial one, evaluated on the entire reference set, tries to reduce the set of candidates as much as possible, such that the computationally more expensive steps have to be evaluated only on a minimal sub-set of the initial one. For example, the localization at different levels of accuracy with omnidirectional images was previously proposed by Gaspar *et al.* [13]. Their navigation method consists of two steps: a fast but less accurate topological step, useful in long-term navigation, followed by a more accurate tracking-based step for short distance navigation.

One of the advantages of our hierarchical method is that it achieves accurate metric localization with a minimal number of nearby reference images. Most of the previously proposed vision-based approaches for global localization do not go further than the scene identification. An example is [4], where the performance in scene recognition using image global descriptors, histograms, is compared against local features, SIFT. In other approaches that aim at metric localization, the location of one of the images from the VM is given as the final localization result. This implies that accurate results can only be achieved with a relatively high density in the stored image set. For example in [14], a hierarchical localization method for omnidirectional images based on the Fourier signature is proposed. It returns a location area around the selected reference images from only one environment. It is computationally efficient but the density of reference images stored directly influences the accuracy obtained in the localization. Nevertheless, with the metric localization we propose, the stored images density is only restricted by the wide-baseline matching that the local features are able to deal with. Recently in [2], the scene recognition was followed by a proposal for two view metric localization estimation, with some promising results. However, in our work several substantial changes are proposed, both in the scene recognition with a different method, and in the metric localization achievement through three bearing-only data views. The fact of using three views of bearing only information in the last step gives additional interesting properties to this approach, besides solving the structure

of the scene. Matches in three views are more robust than two view ones, and moreover, since two reference views are used the scale and multiplicity of solutions problems are directly solved.

The image features used in this work are vertical scene lines, which are projected in omnidirectional images as radial ones (supposing vertical camera axis). They are quite suitable for our needs: to be able to deal with wide baselines, illumination changes and occlusions to an acceptable extent, without increasing too much the execution time of the method. The number of features proposed for matching has increased largely over the last few years, where SIFT [15] has become very popular. Also many modifications of this feature have been proposed, with some simplifications to improve the performance for real time applications, e.g. in [16] or [17]. Yet, in this work we use the radial lines because of their interesting advantages, especially when working with omnidirectional images.

The radial lines together with their descriptors are detailed in Section 2. Afterwards, all the steps of the localization method are explained in Section 3. Finally, in the experimental results Section 4 we evaluate the performance of the developed method, using the proposed radial lines and compare it with results obtained by using a state-of-the-art wide-baseline matching technique, namely the scale invariant features (SIFT) extractor provided by D. Lowe [15]. The results show significantly lower computational cost using our radial lines without too much loss in performance and as such prove the effectiveness of the different steps of the hierarchical method.

2 Lines and their Descriptors

In this proposal, the features used are scene vertical lines with their image support regions. As mentioned before, these lines show several advantages, specially when working with omnidirectional images. They can be extracted and processed fast and they represent interesting natural landmarks in the scene, such as doors or walls. Moreover, they allow us to automatically estimate the center of projection in omnidirectional images and are less sensitive to optical distortions in that kind of images (every vertical in the scene will be projected as a radial one in the image).

Each line should be described by a set of descriptors that characterize it in the most discriminant way possible, although it is necessary to find a balance between invariance and discriminative power, as the more invariant the descriptors are the less discriminant they become. In this section, we first explain the line extraction process (Section 2.1), whereafter we shed light on the kind of descriptors proposed to characterize them (Section 2.2).

2.1 Line Extraction

The line features and their Line Support Region (LSR) are extracted using our implementation of the algorithm [18]. Firstly, this extraction algorithm computes the image brightness gradient, and it segments the image into line support regions, which consist of adjacent pixels with similar gradient direction and gradient magnitude higher than a threshold. Secondly a line is fitted to each of those regions using a model of brightness variation over the LSR [19]. Some LSR's with their fitted line are shown in Fig. 1.

As mentioned before, only vertical scene lines are used. The optical axis of the camera is supposed to be perpendicular to the floor and the motion is assumed to be planar. Under these conditions, those lines are the only ones that keep always their straightness, being projected as radial lines in the image. Therefore, they are quite easy to find and we can automatically obtain from them the center of projection in the omnidirectional image, an important parameter to calibrate this kind of images. It is estimated with a simple *ransac*-based algorithm that checks where the radial lines are pointing. As it can be seen on the right of Fig. 1, the center of projection is not coincident with the image center. The more accurate we find its coordinates the better we can estimate later the multi-view geometry and therefore, the robot localization.

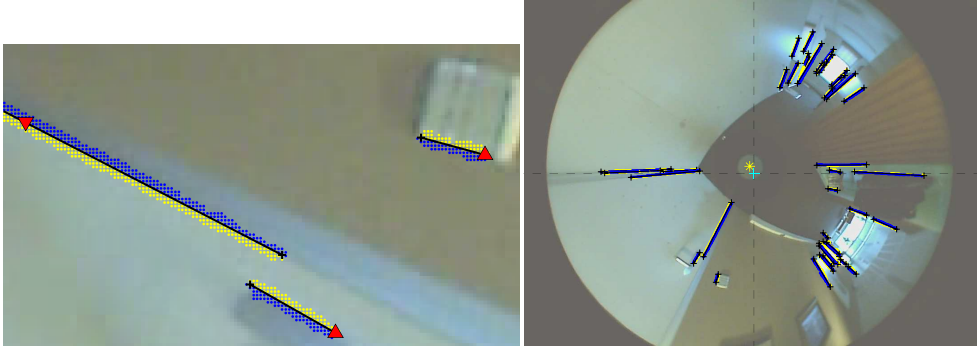


Fig. 1. Left: detail of some LSRs, divided by their line (the darkest side on the right in blue and the lighter on the left in yellow). A red triangle shows the lines direction. Right: all the radial lines with their LSR extracted in one image. The estimated center of projection is pointed with a yellow star (*) and the image center with a blue cross (+).

2.2 Line Descriptors

We propose to use the following descriptors to characterize the radial lines features. All descriptors except the last group (*geometric descriptors*) will be

computed separately over each sides in which the LSR is divided by its line (Fig.1).

- *Color Descriptors.* We have worked with three of the color invariants suggested in [20], based on combinations of the generalized color moments,

$$M_{pq}^{abc} = \int \int_{LSR} x^p y^q [R(x, y)]^a [G(x, y)]^b [B(x, y)]^c dx dy. \quad (1)$$

being M_{pq}^{abc} a generalized color moment of order $p + q$ and degree $a + b + c$ and $R(x, y)$, $G(x, y)$ and $B(x, y)$ the intensities of the pixel (x, y) in each RGB color band centered around its mean. These invariants are grouped in several classes, depending on the scheme chosen to model the photometric transformations. In order to keep a compromise between complexity and discriminative power, we chose as most suitable for us those invariants defined for scale photometric transformations using relations between couples of color bands. The definitions of the 3 descriptors chosen are as follows:

$$DS^{RG} = \frac{M_{00}^{110} M_{00}^{000}}{M_{00}^{100} M_{00}^{010}} \quad DS^{RB} = \frac{M_{00}^{101} M_{00}^{000}}{M_{00}^{100} M_{00}^{001}} \quad DS^{GB} = \frac{M_{00}^{011} M_{00}^{000}}{M_{00}^{010} M_{00}^{001}} \quad (2)$$

- *Intensity Frequency Descriptors.* We also use as descriptors the first seven coefficients of the Discrete Cosine Transform, DCT , over the intensity signal (I) along the LSR of each line. The DCT is a well known transform in the areas of signal processing [21] and widely used for image compression. It is possible to estimate the number of coefficients necessary to describe a certain percentage of the image content. For example with our test images, seven coefficients are necessary on average to represent 99% of the intensity signal over a LSR.
- *Geometric Descriptors.* Finally two geometry-related parameters are obtained from the lines. Firstly, the line orientation θ in the image (in 2π range). The second geometric parameter is the line direction δ , a boolean indicating if the line is pointing to the center or not. This direction is established depending on which side of the line is the darkest region.

From the options studied, the following 22 descriptors are proposed to describe the lines: 3 color descriptors (DS^{RG} , DS^{RB} , DS^{GB}) and 7 frequency descriptors ($DCT_1, DCT_2 \dots DCT_7$) computed on each side of the LSR, and 2 geometric properties (orientation θ and direction δ). Notice that to deal with big amounts of images, it would be convenient to decrease this number. Then, in the second step of the proposal (similarity evaluation), which works with the features from a big amount of reference images, a reduced 12-descriptors set will be used: the same as described above but using only the two first DCT components (DCT_1, DCT_2). The descriptors explained were chosen after extensive empirical tests, taking into account some initial matching results and the weights given to each descriptor in a Principal Component Analysis (PCA).

3 Hierarchical Localization Method

Our proposal consists of a hierarchical vision-based global localization performed in three steps. With this kind of localization, we can deal with large databases of images in an acceptable time. This is possible because we avoid evaluating the entire data set of images in the most computationally expensive steps. Note also that the VM can store already the extracted image features and their descriptors. Even the matches between adjacent images can be also precomputed. Fig. 2 shows a scheme of the three steps of the method, which are detailed later in this section.

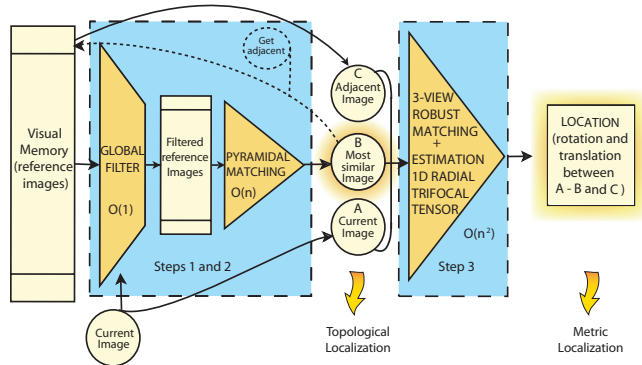


Fig. 2. Diagram of the three steps of the hierarchical localization.

As mentioned previously, the time complexity increases in each step of the method with regard to the previous ones:

- The first step (Section 3.1) has constant complexity with the number of features in the image. Here, all the images in the VM are evaluated.
- The second step algorithm (Section 3.2) has linear complexity with the number of features. Only the images that passed the previous filter step are evaluated here, using 12 descriptors per feature.
- The third step process (Section 3.3) has quadratic complexity with the number of features, and a 22-element descriptor vector per feature is used. However, it evaluates only the query image and two images from the VM. Indeed, in each step we have a computational cost linear with the number of images remaining on it.

3.1 Global Descriptor Filter

In this first step, three color invariant descriptors are computed over all the pixels in the image. The descriptor used is DS , described previously in Section 2.2, that is computed for each possible pair of RGB bands. All the images in the VM are compared with the current one, with regard to those 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 colors in the image.

3.2 Pyramidal Matching

This step finds the image which is the most similar to the current one. For this purpose, the set of descriptors of each line is used to implement a *pyramid matching kernel* [7]. This consists of building for each image several multi-dimensional histograms (each dimension corresponds to one descriptor), where each line feature occupies one of the histogram bins. The value of each line descriptor is rounded to the histogram resolution, which gives a set of coordinates that indicates the bin corresponding to that line.

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 one of the visual memory (v), is obtained by finding the intersection of the two pyramids of histograms:

$$S(\psi(c), \psi(v)) = \sum_{i=0}^L w_i N_i(\psi(c), \psi(v)) , \quad (3)$$

with N_i the number of matches (LSRs that fall in the same bin of the histograms, see Fig. 3) 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 *normalized* distance obtained is

$$S_{cv} = \frac{S(\psi(c), \psi(v))}{\sqrt{S(\psi(c), \psi(c)) S(\psi(v), \psi(v))}} . \quad (4)$$

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).

Once the similarity measure between our actual image and the VM images is obtained, we choose the one with highest S_{cv} . Based on the annotations of this chosen image, the robot identifies the current room.

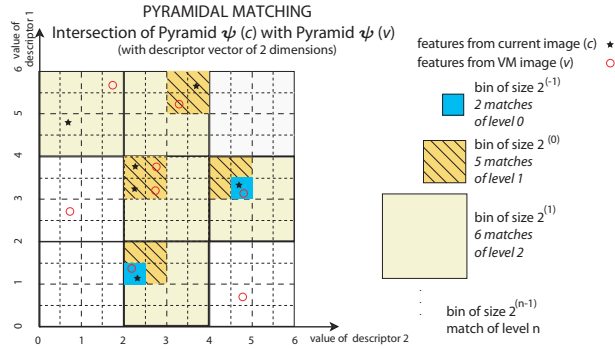


Fig. 3. Example of Pyramidal Matching, with correspondences in level 0, 1 and 2. For graphic simplification, with a descriptor of 2 dimensions.

3.3 Line Matching and Metric Localization

This section details the third step of the hierarchical method, that will provide a more accurate localization information if needed. As mentioned before, from previous steps the matches are not all feature-to-feature. Besides, the method used there is likely to provide outliers as it is based only in appearance. At this point, individual matches are required to simultaneously estimate a robust set of three view matches and the 1D trifocal tensor and afterwards, the localization from it. Once we obtain the most similar image from previous step, we find two-view line matches between the current and the most similar image, and between that most similar and one adjacent in the VM (subsection 3.3.1). Afterwards, these two view matches will be extended to three views in order to estimate the 1D tensor (subsection 3.3.2).

3.3.1 Line matching algorithm.

The two-view line matching algorithm we propose works as follows:

- Firstly, decide which lines are *compatible* with each other. They must have the same direction (δ). Besides, the relative rotation between each line and one compatible in the other view has to be consistent, i.e., it must be similar to the global rotation between both images obtained with the average gradient of all image pixels. As explained, the rest of descriptors are computed in regions along the line in both sides. Then to finally classify two lines as compatible, at least the descriptor distances on one side of the LSR should be below the established threshold.
- Second, to compute a *unique distance* is necessary. This is done using all the distances from color and intensity frequency descriptors between each pair of compatible lines. Instead of using the classical Mahalanobis distance, which needs a lot of training to compute the required covariance matrix with

satisfying accuracy, we just normalize those distances between 0 and 1. The purpose is to have the distances of the different kinds of descriptors in similar scales, to be able to sum them. The simple normalization used works well in practice and it is more adaptable to different queries. A correction or penalty is also applied to increase the distance with the ratio of descriptors (N_d) whose differences were over their corresponding threshold in the compatibility test. So the final distance considered between two lines, i and j , will be

$$d_{ij} = (d_{RGB}^{Min} + d_{DCT}^{Min})(1 + N_d), \quad (5)$$

where d_{RGB}^{Min} and d_{DCT}^{Min} are the smallest distances (in color and intensity descriptors respectively) of the two LSR sides.

- Then, a *nearest neighbour matching* between compatible lines is performed.
- Afterwards, we apply a *topological filter* to the matches, to help to reject non-consistent ones. An adaptation of the filter proposed in [22] is done for radial lines in omnidirectional images. It is based on the probability that two lines will keep their relative position in both views. It improves the robustness of this initial matching, although it can reject some good matches. However, those false negatives can be recovered afterwards and robustness is more important in this step.
- Finally, we run a *re-matching* step, that takes into account the fact that the neighbours to a certain matched line should rotate from one view to the other in a similar way.

3.3.2 Metric Localization using the 1D Radial Trifocal Tensor.

It is well known that to solve the structure and motion problem from lines at least three views are necessary [23]. After computing two-view matches between the two couples of images explained before, the matches are extended to three views intersecting both two-view sets. These trios of corresponding features compose what we call *Basic matching* set.

A robust method (*ransac* in our case) is applied to the three-view matches to simultaneously reject outliers from the *Basic matching* and estimate a 1D radial trifocal tensor. The orientation (θ) of each line \mathbf{r} in the image is expressed by 1D homogeneous coordinates ($\mathbf{r} = [\sin \theta, \cos \theta]$). The trilinear constraint, imposed by a trifocal tensor on the projection coordinates of a line \mathbf{v} in three views ($\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$), is defined as follows

$$\sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^2 T_{ijk} \mathbf{r}_{1(i)} \mathbf{r}_{2(j)} \mathbf{r}_{3(k)} = 0, \quad (6)$$

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} . Fig. 4 shows a vertical line

projected in the three views and the location parameters to estimate.

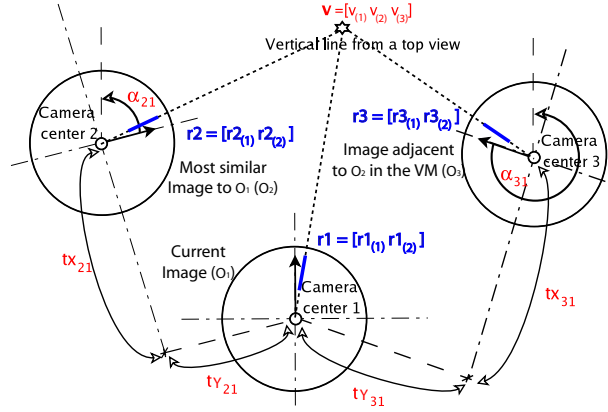


Fig. 4. Projection of a landmark \mathbf{v} in three views and the parameters of the second and third location relative to the first: rotations, α_{21} and α_{31} , and translations, $[t_{x21}, t_{y21}]$ and $[t_{x31}, t_{y31}]$.

With five matches and two additional constraints defined for the calibrated situation (internal parameters of the camera are known), we have enough to compute a 1D trifocal tensor [9]. In case of omnidirectional cameras, the 1D radial trifocal tensor is used and only the center of projection has to be calibrated. From this tensor, we can get a robust set of matches, the camera motion and the structure of the scene [8]. The results obtained with that method are estimated up to a scale and with a double-solution ambiguity, but both issues can be solved easily with the *a priori* knowledge we have (the relative position of the two images in the VM).

4 Experiments and Results

In this section, the performance of the proposed algorithms is shown, both for recognizing the current room (4.1) and for metric localization (4.2).

Two **data sets** have been used: *Almere* and our own (named *data set LV*). In Fig. 5 there is a scheme of the rooms from both databases, the second one with details of the relative displacements between views. All images have been acquired with omnidirectional vision sensors with hyperbolic mirror.

- *Data set LV* consists of 70 omnidirectional images (640x480 pixels). From them, 37 are classified, sorted in different rooms (from 6 to 15 images per room depending on the size of the location). The rest corresponds to unclassified ones from other rooms, other buildings or outdoors. All images from this data set composed the visual memory VM_{LV} used in the experiments.

- *Almere* data set was provided for the workshop [24], as well with the images sorted in different rooms. In the presented experiments, we used the low quality

videos (about 2000 frames per round) given there, from rounds 1 and 4 of a robot moving around a typical house environment. These rounds were obtained in dynamic environments, with people walking around. Only every 5 frames were extracted for the experiments, half for reference (the even frames: *fr10-fr20-fr30-...*) and the other half for testing (the odd frames: *fr5-fr15-fr25-...*). Taking only one fifth of the frames from round-1 selected for reference (*fr50-fr100-fr150-...*), another visual memory (VM_1) was built for the experiments.

- Finally, both data sets VM_{LV} and VM_1 were joined, composing a bigger visual memory, VM_{ALL} , to make more exhaustive tests about the robustness of the method.

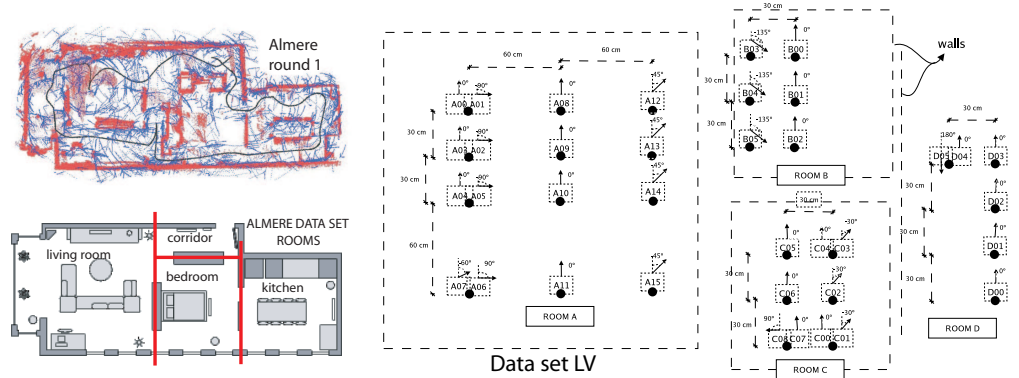


Fig. 5. Schemes of the rooms from data sets used in the experiments. Left: *data set Almere*. Right: *Data set LV* with the locations of the views that compose it.

The timing information given in some of the tests intends only to allow some efficiency comparisons. It should be taken into account that all tests were run in Matlab, and the implementations were not optimized for speed (the complexity of all the steps was analyzed previously in Section 3). Besides, there is no pre-computation performed for any of the steps that could afford it, e.g the feature extraction and matching between the images in the VM, what would be done for real-time executions.

4.1 Topological Localization: finding the current room.

In this section, experiments of the two first steps of the method are shown to evaluate the topological localization. Exhaustive tests with query images from *Almere* round-1 (*Almere-1*) and round-4 (*Almere-4*) and *data set LV* were performed. Every classified image in *data set LV* was compared with the rest of the VM_{LV} (i.e. all the possible 37 tests), and to execute a similar number of tests for *Almere* data, we took 1 fifth of the frames selected from there for testing, both from (*Almere-1*) and (*Almere-4*).

First step

The purpose of this first step is a pre-filtering of the reference images, to reject quickly those which look very different from the current query. After different tests with all the data available, a threshold of 60% was established for the difference allowed in the three color global descriptors comparisons (explained in Section 3.1). Every reference image with higher difference than that threshold is already discarded at this point.

Indeed this pre-filtering should have the minimum amount of false negatives possible (reference images rejected which belong to the same room as the query). The performance using queries from *data set LV* compared to reference views from VM_{LV} or queries from *Almere-1* compared to VM_1 was quite similar, with an acceptable average in both cases of less than four false negatives for the threshold chosen (60%). The pre-filter behaviour in one of these cases using different criteria (threshold) is shown on the left of Fig. 6. The performance was indeed worse in a third more difficult case, where queries from *Almere-4* were used with reference images from VM_{ALL} . Most images original from VM_{LV} were properly rejected in this step, however the false negative ratio increased. This is because of the important differences in the global appearance of the images, due to many occlusions and moving people in *Almere-4*. The performance in this case, using query images from *Almere-4*, of the pre-filtering with different threshold criteria is shown on the right of Fig. 6. Global image descriptors must be carefully used because they show no robustness against occlusions, and for example they could reject all possible correct reference images.

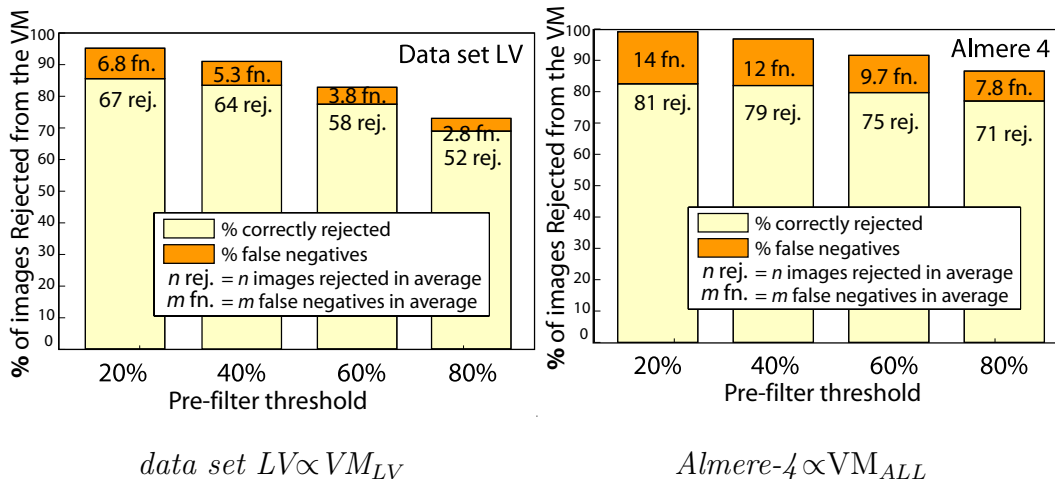


Fig. 6. Images rejected and false negatives in step 1 for two cases of study.

Second step

The goal of this step is to give the topological localization of the current view. In order to achieve that, a similarity evaluation was run with the images that passed the previous pre-filtering step. The most similar image found was considered a correct localization if it belonged to the same room as they query. This step was evaluated using the proposed line features as well as SIFT, which can be considered the state of the art in local feature matching, using the implementation provided by D. Lowe [15]. Firstly in this step, note that to build the pyramid of histograms, a suitable scaling and/or normalization of the feature descriptors is required. The discrete size of the pyramid bins makes necessary to have all the descriptors in the same range and well distributed.

Pyramidal vs. Individual Matching to get a similarity score.

To better evaluate this second step, its performance was compared with the same task done using individual matching. The results of the algorithm developed for matching lines in Section 3.3 could also be used for searching the most similar image. For this purpose, we can compute a similarity score that depends on the matches found (n) between the pair of images, weighted by the distance (d) between the lines matched. It has to be taken 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

$$DIS = n d + F_1(1 - P_o) + F_2(1 - P_o) . \quad (7)$$

Results for both methods in the room recognition tests are shown in Table 1, for the three cases studied and using radial lines or SIFT features. A test was considered correct if the image chosen as most similar was from the correct room.

Table 1

Topological localization (room recognition) using VM_{LV} and VM_1 . *Pyr*: Correct tests with the Pyramidal matching similarity score. *IM*: Correct tests with the individual matching similarity score. T_{Pyr}, T_{IM} : execution time for each comparison of a query image with one from the reference set for each method respectively.

	Almere1 \propto VM ₁				Almere4 \propto VM ₁				Data LV \propto VM _{LV}			
	Pyr	IM	T_{Pyr}	T_{IM}	Pyr	IM	T_{Pyr}	T_{IM}	Pyr	IM	T_{Pyr}	T_{IM}
lines	90%	80%	0.35 s	0.95 s	52%	50%	0.35 s	0.9 s	82%	93%	0.1 s	0.3 s
sift	78%	85%	130 s	35 s	60%	70%	140 s	36 s	45%	95%	38 s	8.7 s

In case of using radial lines as image features, the correctness in the results was similar for both similarity evaluation methods. However, there is a clear advantage in efficiency for the Pyramidal matching, that takes around one third

of time than the individual matching. On the other hand, the correctness in the results for SIFT was also similar with both similarity evaluations methods, but much faster with the individual matching. The worse performance of SIFT using the Pyramidal matching could be expected, as that method is not suitable for features descriptors as long as the SIFT one. In this cases the dimension of the histograms grows a lot, then too much time is required by this method in the computations and to find correspondences in the bins of the Pyramids. Therefore, the correctness in results was good for both features, with a little advantage for SIFT when the data used was more challenging, but with much higher computational cost than the radial lines. Notice that it takes similar time to compare one query with one reference image using SIFT with its best performance (35 s.) than to compare one query with the whole VM using radial lines (0.35 s. * 40 reference views).

In the tests $Almere4 \propto VM_1$ the performance was always lower than in the other cases, but we should take into account that the queries belong to *Almere4*, a video recorded with many occlusions, while the reference set (*Almere1*) is from a "cleaner" round. This makes the first step work worse and the global appearance based descriptors less useful. This confirms the importance of using local features to deal with occlusions. If we skip the first step that is based only on global appearance, the performance increases but also the computation time will grow quite a lot.

Some other experiments were performed to test the robustness of the process using VM_{ALL} , the biggest VM available. The experiments shown before were repeated using this VM_{ALL} , and the new results are shown in Table 2. The results there pointed out that increasing the number of possible rooms and the VM size does not reduce the classification rates too much.

Table 2

Topological localization (room recognition) using VM_{ALL} . *Pyr*: Correct tests with the Pyramidal matching similarity score. *IM*: Correct tests with the individual matching similarity score. $T_{Pyr}|T_{IM}$: execution time for each comparison of a query image with one from the reference set for each method.

	Almere1 $\propto VM_{ALL}$				Almere4 $\propto VM_{ALL}$				Data set LV $\propto VM_{ALL}$			
	Pyr	IM	T_{Pyr}	T_{IM}	Pyr	IM	T_{Pyr}	T_{IM}	Pyr	IM	T_{Pyr}	T_{IM}
lines	80%	70%	0.18 s	0.2 s.	50%	63%	0.1 s	0.15 s.	86%	90%	0.14 s	0.3 s.
sift	70%	90%	92 s	9 s.	60%	70%	64 s	6 s.	75%	86%	50 s	5 s.

Also the behaviour of the method when the size of the VM is decreased (to half, one third, one quarter,...) has been studied, and the results showed the method to be quite robust in this aspect too. As long as there is still some correct reference image in the VM and the density of images per room decreases in a similar way for all rooms, the performance remains similar. Note that when the VM is reduced, the possible correct reference images are reduced, but also

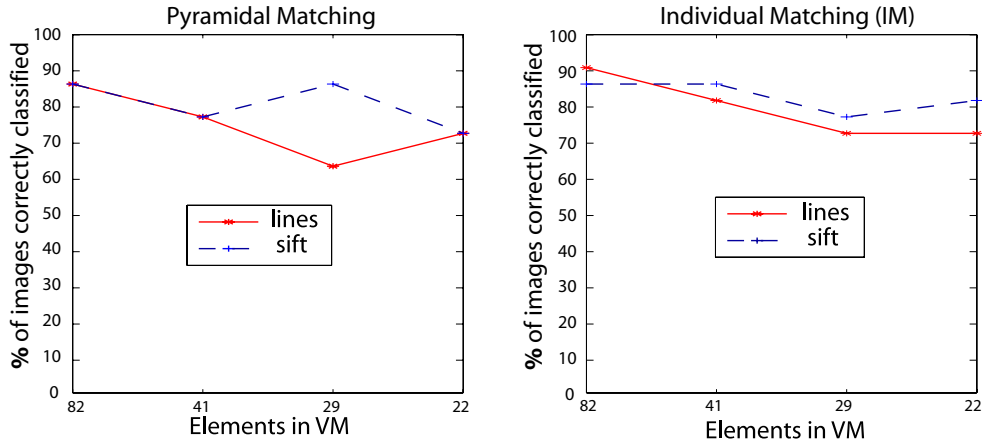


Fig. 7. *Data set* $LV \propto VM_{ALL}$. Evolution of the performance in room identification when the density of reference images in the VM decreases. Right: using Individual matching. Left: using the Pyramidal matching.

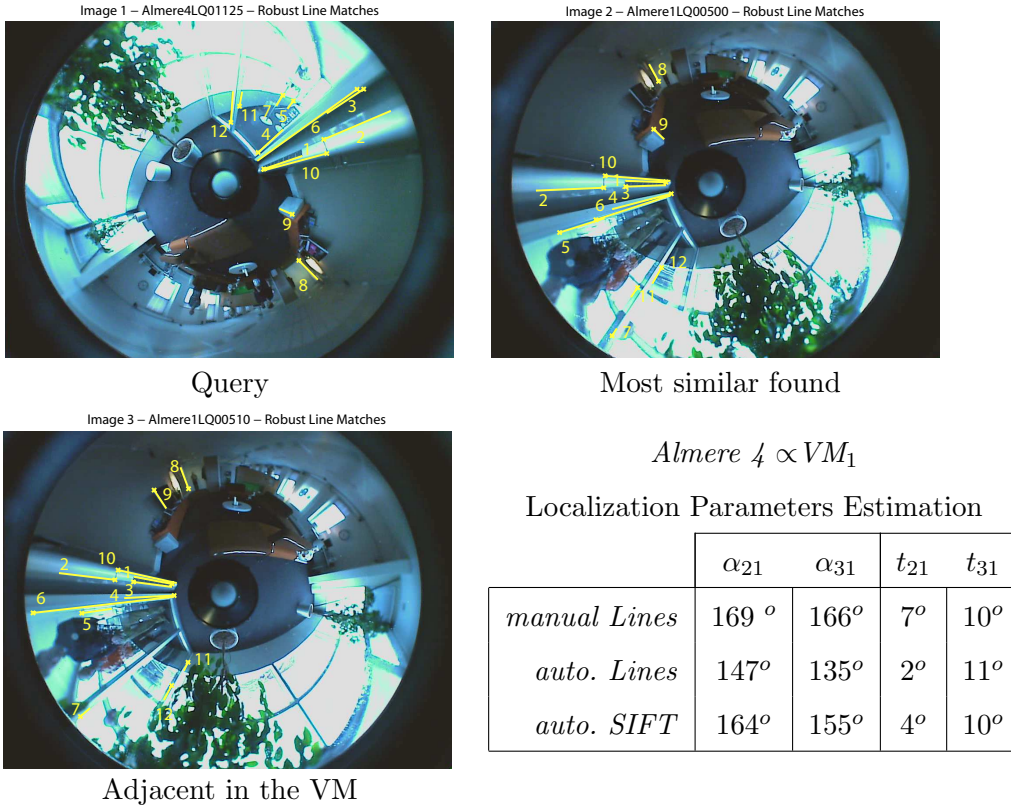
the images that could cause misclassifications. Fig. 7 represents the results of the method when the VM size decreases, with the percentage of correct room identification tests using lines and SIFT. Only one of the cases of study considered is shown here, as the behaviour was quite similar in all of them.

4.2 Metric Localization Results

As explained before, we need three views to get the metric localization. Then, trios of correspondences are obtained from the two-view matches between the current image and the most similar selected, and between this one and one of its neighbors in the VM. The 1D radial trifocal tensor is robustly estimated from those trios, and simultaneously a robust set of three-view matches is established (those who fit the geometry of the tensor). Fig. 8 shows an example of the robust line matches and localization parameters (rotation and translation direction) obtained after applying the whole method to a query image from Almere-4 using the VM_1 . Ground truth was not available for this experiment, then results using SIFT and manual line matching are shown to compare with the results obtained using our automatic line matching. In this test, we can see a clear example of the advantage of the method used to obtain the metric localization, with regard to methods that give as localization the location of the most similar image. In this case the most similar image is rotated around 160° in relation to the query. That would be the rotation estimation error if the localization would be determined by the most similar image location. In this situation, of a more challenging matching case, SIFT seems to behave a little better than the radial lines.

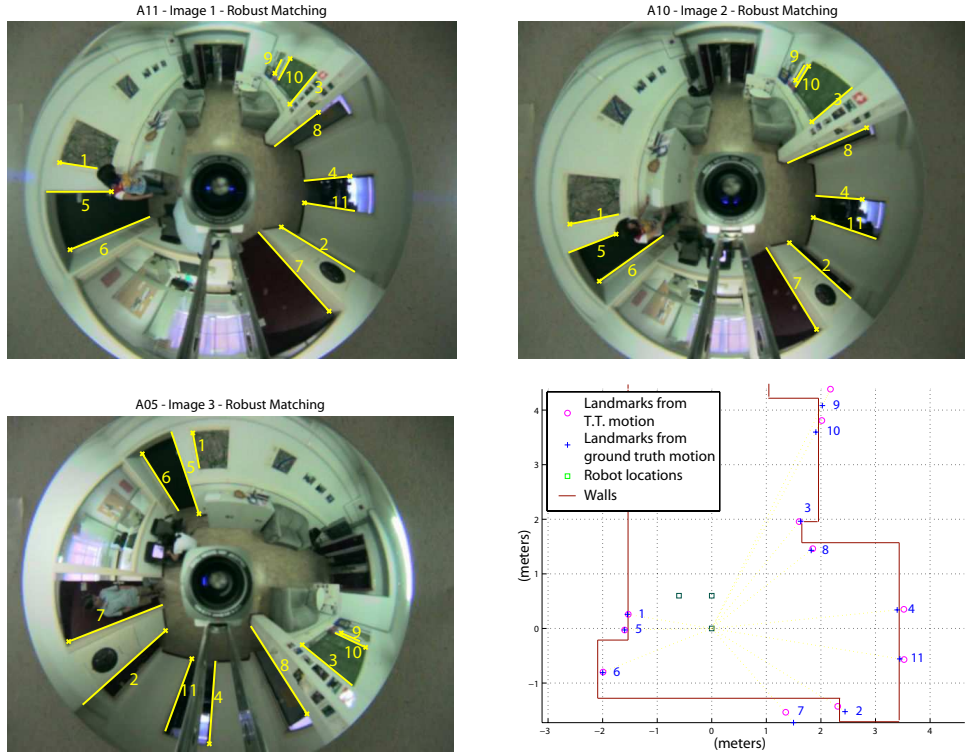
Detailed ground truth was only available for *data set* LV , therefore a little more

detailed evaluation of this metric localization was done with data from there. In Fig. 9 there is an example showing robust matches together with camera and landmarks location recovery information. Notice the small average errors, specially if we take into account the uncertainty of our ground-truth (obtained with a metric tape and goniometer), as well as the consistence of the results showing quite small standard deviations. Radial lines seem quite suitable for the scene reconstruction, because they give less number of matches than SIFT but they correspond with significant objects in the scene (such as corners or doors). Moreover, the radial lines seem to have a more accurate orientation in the image, giving better reconstruction results. Probably, the best option for the metric localization would be to use a *mixed* solution, that takes the advantages of both lines and SIFT.



Basic matches: 15 (5 mismatches). Robust matches: 12 (3 mismatches).

Fig. 8. Example of triple robust matches obtained between the current position or query (first image) and two images from the VM (second and third images). Table with rotation and translation direction between each of the two reference images and the query one. Results obtained with manual line matching (*manual Lines*), with automatic line matching (*auto. Lines*) and with SIFT (*auto. SIFT*).



Basic matches: 17 (6 mismatches). Robust matches: 11, all correct.

	Localization errors				Landmarks
	α_{21} (std)	α_{31} (std)	t_{21} (std)	t_{31} (std)	reconstr. err (std)
lines	0.9°(0.1)	2°(0.07)	0.8°(0.2)	7°(0.3)	0.18m. (0.01)
sift	0.8°(0.002)	1.8°(0.01)	2.5°(0.01)	7.8°(0.44)	0.22m. (0.05)

Fig. 9. *Experiment 2R*. Top: Omnidirectional images and robust line matches estimated with the tensor, and scheme of scene reconstruction obtained through the tensor computed with line matches (pink o) and ground truth (blue +). Bottom: tables with robot and landmarks localization errors.

5 Conclusions

In this work, a new efficient vision based method for global localization has been proposed. Our three-step approach uses omnidirectional images and combines topological and metric localization information. The localization task computes the position of the robot relative to a set of reference images from different rooms. After a rough selection of a number of candidate images based on a global color descriptor, the best resembling image is chosen based on a pyramidal matching with wide baseline local features based on radial lines. This kind of feature enables the grid of reference images to be less dense as well as improves the behaviour against occlusions, illumination changes or noise. The image chosen as most similar provides the topological localization

(current room). A third step involving the computation of the omnidirectional trifocal tensor yields the metrical coordinates of the unknown robot position. As a result, our approach provides accurate localization with a minimal reference data set, contrary to the approaches that give as current localization the location of one of the reference views. The experimental results on two different data sets of omnidirectional images show the efficiency and high accuracy of the proposed method.

References

- [1] David C.K. Yuen and Bruce A. MacDonald. Vision-based localization algorithm based on landmark matching, triangulation, reconstruction and comparison. *IEEE Trans. on Robotics*, 21(2):217–226, 2005.
- [2] 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.
- [3] S. Se, D. G. Lowe, and J. J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364–375, 2005.
- [4] J. Košecká and F. Li. Vision based topological Markov localization. In *IEEE Int. Conf. on Robotics and Automation*, pages 1481–1486, 2004.
- [5] 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.
- [6] O. Booij, Z. Zivkovic, and B. Kröse. Sparse appearance based modeling for robot localization. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 1510–1515, 2006.
- [7] 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.
- [8] 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 *Proc of the IEEE Int. Conf. on Robotics and Automation*, pages 551–556, 2006.
- [9] 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.
- [10] 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.

- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. Journal of Computer Vision*, 60(2):91–110, 2004.
- [16] L. Ledwich and S. Williams. Reduced sift features for image retrieval and indoor localization. In *Australian Conference on Robotics and Automation*, 2004.
- [17] H. Andreasson, A. Treptow, and T. Duckett. Localization for mobile robots using panoramic vision, local features and particle filter. In *IEEE Int. Conf. on Robotics and Automation*, pages 3343–3353, 2005.
- [18] J.B. Burns, A.R. Hanson, and E.M. Riseman. Extracting straight lines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(4):425–455, 1986.
- [19] J. J. Guerrero and C. Sagüés. Camera motion from brightness on lines. combination of features and normal flow. *Pattern Recognition*, 32(2):203–216, 1999.
- [20] F. Mindru, T. Tuytelaars, L. Van Gool, and T. Moons. Moment invariants for recognition under changing viewpoint and illumination. *Computer Vision and Image Understanding*, 94(1-3):3–27, 2004.
- [21] K.R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, Boston, 1990.
- [22] H. Bay, V. Ferrari, and L. Van Gool. Wide-baseline stereo matching with line segments. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 329–336, 2005.
- [23] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2000.
- [24] Data set from FS2HSC IEEE/RSJ IROS 2006 Workshop. <http://staff.science.uva.nl/~zivkovic/fs2hsc/dataset.html>.