

Incremental Spectral Clustering and Seasons: Appearance-Based Localization in Outdoor Environments

Christoffer Valgren and Achim Lilienthal

Abstract—The problem of appearance-based mapping and navigation in outdoor environments is far from trivial. In this paper, an appearance-based topological map, covering a large, mixed indoor and outdoor environment, is built incrementally by using panoramic images. The map is based on image similarity, so that the resulting segmentation of the world corresponds closely to the human concept of a place. Using high-resolution images and the epipolar constraint, the resulting map is shown to be very suitable for localization, even when the environment has undergone seasonal changes.

I. INTRODUCTION

Appearance-based approaches to topological mapping and localization for mobile robots have been very successful in indoor environments. Both localization and loop closing detection can be handled by robust image matching methods. Methods based on stable local features, such as the Scale-Invariant Feature Transform (SIFT) by Lowe [11], have gained popularity because of the discriminative nature of the features and their invariance to illumination, scale, translation and rotation. A high resistance to occlusions can also be achieved [2].

For smaller environments, it might be sufficient to consider every image as a topological node. Global localization within the map, based on comparing images using local features, is then clearly an $O(n)$ operation. While perhaps acceptable for small maps, as the map grows this quickly becomes infeasible because of the computational cost associated with image matching algorithms. Also, a map constructed in this manner says nothing about the *structure* of the environment; traversing an area multiple times will result in multiple images representing the same place. Thus, it is necessary to cluster the images together into larger nodes.

How should a node be defined? A commonly used notion is that of “distinctive places” [10], i.e., the nodes are positioned at locations that reaches a maximum of “distinctiveness”. As an example of this, consider the method used by Tapus and Siegwart [14]. They extract features (edges, corners, colour patches) from omnidirectional images and combine them together with laser scans into “fingerprints”, which acts as nodes in a topological map. Another approach is used by Booi et al. [5]; every image is viewed as a node in a graph, and large clusters are generated by using a

graph partitioning technique called *spectral clustering*. Spectral clustering groups data points based on some similarity measure; thus, the resulting segmentation is close to the idea of distinctive places.

In previous work [16], we presented the *incremental spectral clustering (ISC)* algorithm. It was shown that this algorithm could be successfully applied to the problem of appearance-based topological mapping, generating topological maps in a fraction of the time required by using standard spectral clustering. The natural segmentation of the environment from spectral clustering is mostly retained.

In other previous work [17], we showed that a new type of local feature algorithm, Speeded-Up Robust Features (SURF) by Bay et al. [3], outperforms SIFT for the localization task in outdoor environments, both in terms of accuracy and speed. Here, we exclusively use a variant of SURF that is not rotational invariant, “upright” SURF (U-SURF).

We conclude this research arc by showing how to correctly localize in large, mixed indoor and outdoor environments, even over large seasonal variations. We also give an algorithm that can construct a topological map of the environment using incremental spectral clustering and show that it is possible to localize efficiently within this map. For additional details on the methods used here, and also for additional results, see [15].

There are many works in relation to appearance-based global localization using local features, for example by Se et al. [13]. However, there are only a few other approaches to topological localization (image matching) using outdoor images from different seasons, perhaps because it is a difficult process that involves data acquisition over time. Zhang and Kosecka [20] concentrate on recognizing buildings in images, using a hierarchical matching scheme where a “localized colour histogram” is used to limit the search in an image database, with a final localization step based on SIFT feature matching. He et al. [8] also use SIFT features, but employ learning over time to find “feature prototypes” that can be used for localization. Recently, Goedeme et al. [7] used color-enhanced SIFT features and Dempster-Shafer theory to avoid false links between different parts of a topological map, built by using omnidirectional images.

II. SPEEDED-UP ROBUST FEATURES

A brief outline of the SURF algorithm is given here for completeness, for details see Bay et al. [3].

The SURF algorithm consists of a detector and a descriptor. The detector is a blob detector, which approximates the determinant of the Hessian of the Gaussian second

This work is partially supported by The Swedish Defence Material Administration.

The authors are with the Centre for Applied Autonomous Sensor Systems, Örebro University, SE-70182 Örebro, Sweden.
christoffer.wahlgren@tech.oru.se,
achim@lilienthals.de

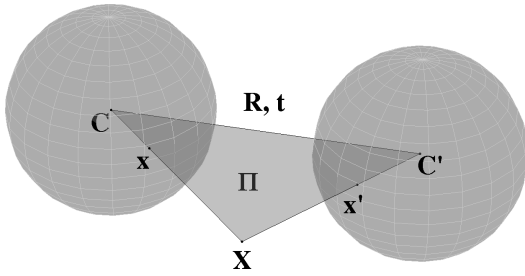


Fig. 1. Epipolar geometry for spherical images.

derivatives by using *box filters*; thus, SURF does not require the explicit computation of scale-space images. Instead, the approximation $D(x, y, \sigma)$ is constructed directly by using an *integral image*. Keypoints (local maxima) are found from the scale space images $D(x, y, \sigma)$ by non-maximum suppression in a $3 \times 3 \times 3$ neighbourhood around each sample point.

To achieve rotation invariance, SURF finds the dominant orientation of the keypoints by utilizing *Haar wavelets*. After rotation of the keypoint neighbourhood, the neighbourhood is divided into 4×4 subregions. The vertical and horizontal Haar wavelet responses are utilized to compute each subregion feature vector of length 4. Stacking the feature vectors for all subregions gives a feature vector for the keypoint of length 64. The final feature vector is obtained by normalization to unit length.

SURF has several descriptor types of varying length. Of interest to this work is in particular “upright” SURF, U-SURF. U-SURF ignores the computation of the dominant orientation and the subsequent rotation of the keypoint neighbourhood. As U-SURF thus leaves out rotation invariance, U-SURF is useful for detecting and describing features in images where the viewpoints are differing by a translation and rotation in the plane (i.e. planar motion).

III. FEATURE MATCHING

A. Matching Local Features

Comparing two images is done by comparing the keypoint descriptors extracted from the images. Depending on the application, there are different matching strategies. A common method, proposed by [11], is to compute the nearest neighbour of a feature descriptor, and check if the second closest neighbour is further away than a threshold value. In this *relative* matching scheme, the nearest neighbour computation is based on the Euclidean distance between the descriptors. To reduce the number of false matches, one can require *reciprocal* matches, i.e. the feature f_i in image i is a match to feature f_j in image j iff f_j is also a match to f_i .

The resulting number of matches will be used later as a measure of similarity between the images.

B. Epipolar Constraint

To further improve the matching result between two spherical images, the *epipolar constraint* can be used to eliminate false matches.

Assume that the positions of the cameras are C and C' (see Figure 1). The first camera C is placed at the origin

and the second camera is translated by t and rotated by the rotation matrix R . The points of correspondence x and x' are both projections on unit spheres of the world point X . A plane Π is formed with the following constraint [4]:

$$x'^T E x = 0 \quad (1)$$

Planar motion can be described with three parameters; two coordinates describing the translation in the ground plane and the rotation θ around an axis perpendicular to the ground plane. Thus, assuming the ground corresponds to the y -plane, the essential matrix E will have the form

$$E = \begin{bmatrix} 0 & -t_z & 0 \\ -t_x \sin\theta - t_z \cos\theta & 0 & t_x \cos\theta - t_z \sin\theta \\ 0 & t_x & 0 \end{bmatrix} \quad (2)$$

Four correspondences are sufficient to obtain the coefficients of this essential matrix [9]. Note that there are in total four solutions; however, only one of the solutions is correct. Here, we choose the following criterion to determine whether two unit vectors x and x' , separated by a unit vector \hat{t} , are pointing in the same, “general” direction.

$$(x \times \hat{t}) \cdot (x' \times \hat{t}) > 0 \text{ and } \arccos(x \cdot \hat{t}) < \arccos(x' \cdot \hat{t}) \quad (3)$$

This relation says that the vectors x and x' along the vector \hat{t} should point towards each other, while the angles between the vectors across the vector \hat{t} should not differ by more than 90° .

Using the epipolar constraint as a model in a sampling algorithm, such as RANSAC [6], it is possible to eliminate false matches and improve matching performance.

C. Matching Images from Different Seasons

In [17], it was shown that it is not possible (with the current method) to use low-resolution panoramic images to perform single-image, global localization in cross-season outdoor environments. However, using U-SURF features extracted from high-resolution images, and also applying the epipolar constraint, the outcome is different.

In Figure 2, top left, the U-SURF feature locations for two low-resolution images are shown. It would be desirable to increase the number of feature matches, so that a more reliable localization can be achieved. However, the low amount of matches between these two images is mainly because the majority of the features are not corresponding to the same objects. Lowering the relative threshold would only lead to more false matches, and worse localization performance.

Another way to increase the number of matches is to utilize high-resolution panoramic images. Because of the relative matching scheme, increasing resolution might not necessarily lead to more matches; however, it is expected that the number of correct matches should increase if the relative threshold is increased. Unfortunately, this also gives more incorrect matches. Many of these false matches can be filtered by using the epipolar constraint, to achieve the final matches shown in Figure 2, bottom left.

This matching scheme is more expensive than using low-resolution images without the epipolar constraint. Excluding feature computation, comparing two low-resolution images requires on average 0.2 s, while comparing two high-resolution images requires on average 1.0 s.

IV. INCREMENTAL SPECTRAL CLUSTERING

A. Spectral Clustering

Spectral clustering does not require that the data can be represented as coordinates in Euclidean n -space – it is sufficient that a similarity measure between the points can be computed. Common for all spectral clustering algorithms is that they take as input an *affinity matrix*, which describes the similarity between the data points. A popular spectral clustering algorithm, used in this paper, is the algorithm by Ng, Jordan and Weiss [12]. The modifications suggested by [18] are also implemented in order to achieve greater numerical stability. For this algorithm, the similarity is based on the distances between the points in the data set influenced by the scaling parameter σ :

$$\mathbf{A}_{ij} = \exp\left(-\left(\frac{d(s_i, s_j)^2}{2\sigma^2}\right)\right) \quad (4)$$

While spectral clustering can be very effective for small data sets, there are issues with these methods:

- The affinity matrix grows with the number of points n as n^2 .
- The number of clusters must be set in advance. This is usually handled by performing the clustering over a varying number of clusters and checking the result after each iteration, or one can use the method described by Zelnik-Manor and Perona [19].
- The entire affinity matrix must be available in order to perform the clustering.

When the affinity matrix entries are costly to compute (for example, if the content is based on the comparison of two high-dimensional vectors), the growing size of the affinity matrix is a serious problem.

B. Incremental Spectral Clustering

The *incremental spectral clustering (ISC)* algorithm avoids the issues outlined above, while being able to produce results similar to the spectral clustering algorithm. Because of its properties, it is very well suited for the task of topological mapping using vision.

The ISC algorithm starts with an empty data set A and thus an empty affinity matrix \mathbf{A} . The method is aptly named; a spectral clustering method is applied to the affinity matrix for every data point added. The algorithm iteratively estimates a *cluster representative* for each cluster. The cluster representative is the data point that is most similar to all other points in the cluster¹.

The cluster representative should not be too dissimilar to any point in the cluster. If it is, the number of clusters must

¹If the data points have a representation in Euclidean space, the cluster representative would be the point closest to the cluster centroid.

Set	Images	Main characteristics
A	139	February. No foliage. Winter, snow-covered ground. Overcast.
B	167	May. Bright green foliage. Bright sun and distinct shadows.
C	944	July. Deep green foliage. Varying cloud cover.
D1	320	October. Early fall, some yellow foliage. Partly bright sun.
D2	234	October. Less leaves on trees, some on the ground. Bright sun.
D3	301	October. Mostly yellow foliage, many leaves on the ground. Overcast.
D4	320	October. Many trees without foliage. Bright setting sun with some high clouds.
Total	2425	

TABLE I

REFERENCE TABLE FOR THE DATA SETS.

be increased and a new clustering is performed. The smallest allowed similarity is called the *similarity threshold*.

Whenever the number of clusters is increased (and when each cluster has a suitable cluster representative), the entries in the affinity matrix that have been assigned to a cluster are replaced by a single cluster representative. The original contents of the cluster are stored for future use in computation of a new cluster representative, if it becomes necessary. The affinity matrix is thus shrunk to a smaller size. The process then continues with the next data point.

ISC requires two external functions:

- A function that computes the affinity between data points. Here, we use the number of matches $M(i, j)$ between two images i and j to compute the corresponding entry in the affinity matrix. The following simple formula for the distance measure² $d(i, j)$ is used in the computation of the affinity (4):

$$d(i, j) = \frac{1}{M(i, j) + 1} \quad (5)$$

- A spectral clustering algorithm that computes k clusters from the current affinity matrix.

Note that ISC is not restricted to the modified NJW algorithm. Any spectral clustering algorithm that takes an affinity matrix and a number of clusters as input could be used without major modifications to the method. For a good overview of various spectral clustering algorithms, see [18]. For further details about ISC, see [16] or [15].

V. DATA SETS

Seven data sets were acquired over a period of nine months, see Figure 2. The data sets span a part of the campus at Örebro University, in both indoor and outdoor locations. The data sets cover different parts of the campus, ranging from 139 up to 944 images (for a total of 2425 images), see Table I.

The images were acquired every few meters; the distance between images varies between the data sets. The data sets do not all cover the same areas. For example, data set D1 does not include the loop around the artificial lake in the northwest corner of the map, see Figure 3.

Data set C, which is also the largest of the data sets, is used as our reference data set (it covers nearly all places visited in the other data sets, see Figure 3).

²Note that $d(i, j)$ is not a true distance measure in the geometric sense. However, it is difficult to construct a true distance metric for image similarity, and the resulting affinity matrix will still be useful.



Fig. 2. **Left, top:** Low-resolution panoramic images matched using U-SURF, relative threshold of 0.65. **Left, center:** High-resolution panoramic images matched using U-SURF, relative threshold of 0.65. **Left, bottom:** High-resolution panoramic images matched using U-SURF, relative threshold of 0.80, reciprocal matching. Epipolar constraint applied. **Center:** Images from northwest part of map. **Right:** Images from parking lot to the south.

The data sets were acquired by an ActivMedia P3-AT robot equipped with a standard consumer-grade SLR digital camera (Canon EOS350D, 8 megapixels) with a curved mirror from 0-360.com. This camera-mirror combination produces omnidirectional images that can be unwrapped into high-resolution panoramic images by a simple polar-to-Cartesian conversion. Each panoramic image is stored in two resolutions, one in full resolution of about 2500×725 , and one in about one third resolution of 800×232 pixels.

To simplify evaluation (cluster visualization and also evaluation of localization result), each image should have a corresponding position. The trajectories of the data sets A and B were determined by hand, while odometry data were available for datasets C, D1, D2, D3 and D4. The odometry was manually corrected, using a Matlab tool specifically designed for this purpose. The resulting trajectories are shown in Figure 3.

VI. EXPERIMENT

In Experiment 1, 40 test images were chosen at random from each data set. These images were then matched against the reference data set C. The number of feature matches between the images was simply taken as a measure of similarity; the image with the highest similarity to the test image was considered to be the winner. Note that this corresponds to global topological localization.

In Experiment 2, the same 40 test images from each data set (as in Experiment 1) were used to perform the localization in a topological map built by the incremental

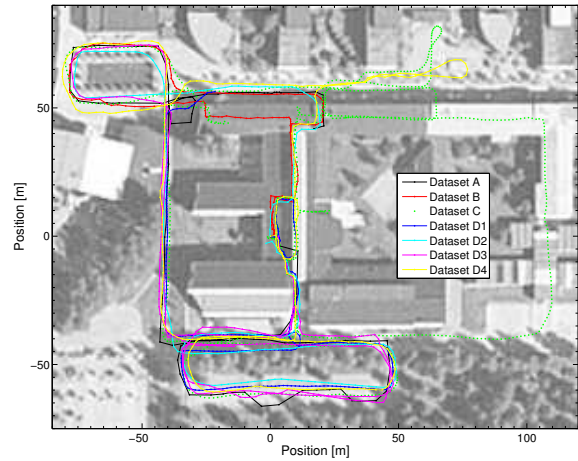


Fig. 3. Ground truth trajectories of all data sets. Note that the parts of data sets B and D4 that are not covered by data set C were excluded from the localization experiments.

spectral clustering algorithm applied to data set C. Each test image is compared to the cluster representatives; all images in nodes with a cluster representative sufficiently similar to the test image are then also compared to the test image.

In both experiments, the localization will be performed by using the matching scheme proposed in Section III-C.

A. Building The Topological Map

When building the topological map, it is not necessary to take seasonal changes within a data set into consideration,

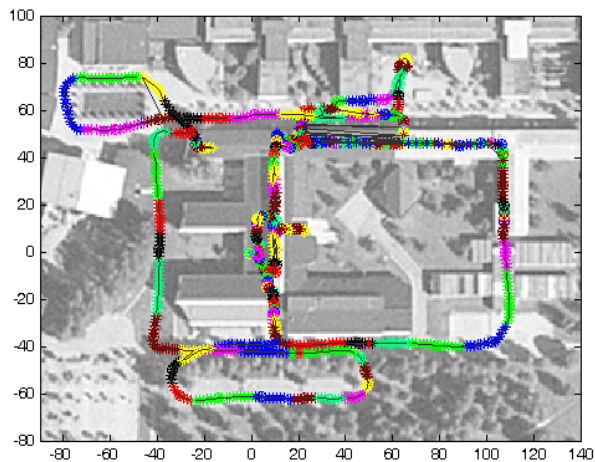


Fig. 4. Topological map produced from data set C. 943 images, 160 nodes.

Data set	Number of comparisons performed	Total number of affinity matrix entries	ISC execution time [s]	Time saved [s]
A	4610	9591	10.7	985.5
B	5891	13861	8.3	1585.7
C	100222	445096	1285.4	67689.4
D1	14440	51040	39.4	7280.6
D2	9061	27261	14.3	3625.7
D3	12575	45150	92.4	6442.6
D4	14144	51040	26.9	7352.3

TABLE II

COMPUTATION TIMES FOR THE TOPOLOGICAL MAPS, EXCLUDING FEATURE EXTRACTION.

and low-resolution images and a simpler matching scheme (U-SURF, relative threshold of 0.7, reciprocal matching) than the one proposed in Section III-C can be used. Figure 4 shows the resulting topological map for reference data set C (the rest of the maps are omitted for space reasons). Each marker corresponds to one image; images belonging to the same node are shown in the same colour (the colours are recycled at regular intervals). All images in a node are also connected by a black line.

Table II lists the total number of image comparisons performed. (The values were obtained by using 100 pairs of images to compute the average comparison time. The mean of this value (0.2 s) was used to compute the total execution time.) This, together with knowledge of the time requirement to do a single comparison, gives an estimate on how much time that was “saved” by doing ISC vs. doing spectral clustering (given that the correct number of clusters was already known).

There are a number of false links in the map from data set C, most notably between two areas of the corridor that are visually very similar. Most of the false links, if not all, could be avoided by introducing corrected odometry (for example from a SLAM scheme such as the one presented in [1]) into the affinity matrix.

Note that the algorithm generates, as expected, more nodes in narrow indoor areas than in larger outdoor areas. This is in particularly clear if one compares the areas around the AASS lab (at the origin) with the long outdoor path 40 meters to the west of the lab. Also, the path nodes correspond roughly to the layout of the building to the east of the path; a major

change in the scene structure (as seen from the moving robot) triggers the creation of a new node. The same pattern can also be seen at the small loop (around an artificial lake) seen in the north-west part of the map.

This observation is interesting for two reasons. First, it shows that ISC successfully can produce clusters that captures structural changes in the environment. Second, there is a certain level of repeatability in the results (which becomes apparent when comparing maps created from different data sets): approximately the same clusters appear even under seasonal variations.

B. Experiment 1: Localization Without a Topological Map

The results from the localization using U-SURF with epipolar constraint are summarized in Figure 5. The high localization rates, even with the simplistic scheme use here, show that it is possible to do localization using local features, even under seasonal variations as in our data sets.

C. Experiment 2: Localization Within the Topological Map

Unfortunately, there is no guarantee that the images chosen as cluster representatives in the map M will be similar to the image I . However, the cluster representatives are the images most similar to all images within the cluster, and the clusters themselves are representations of an area with common features, so there should be a good chance that some similarity exists. If there is no similarity, the localization will obviously fail. The localization procedure is as follows:

- 1) Compare image I with all cluster representatives.
- 2) Select nodes k with more than N_{min} matches.
 - If there is no node that has N_{min} matches or more, choose the node that has the highest match.
- 3) Compare the images in each node k_i with the image I .
- 4) Select the image from the previous step that has the highest number of matches to image I . This is the final localization result.

The localization results using the topological map built from reference data set C is shown in Figure 5. Compared with Experiment 1, the number of correctly localized images is slightly lower. The main reason is that the representative might not be a good match to the image being localized, leading to an incorrect choice of nodes in the localization strategy’s second step. A possible way of avoiding this is to increase the number of nodes in the map, for example by setting the similarity threshold to a higher value. There is a connection between the similarity threshold and the localization rate:

- 1) Setting the similarity threshold to zero will result in a topological map containing only one, large node. The localization rate will be according to Experiment 1.
- 2) Setting the similarity threshold to infinity, will result in every image forming its own node. The localization rate will again be according to Experiment 1!

All values of the similarity threshold between zero and infinity will thus result in some localization rate that (most likely)

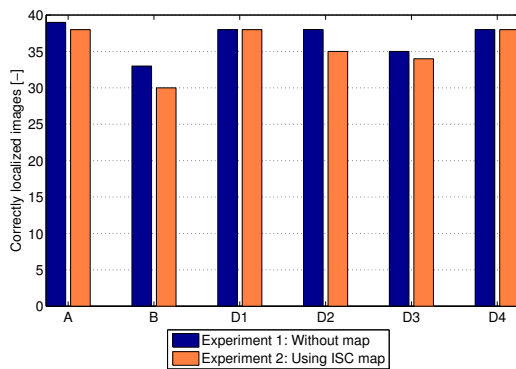


Fig. 5. 40 random images from each data set were localized with respect to the reference data set C, using U-SURF with a relative threshold of 0.9.

will be lower than the values of Experiment 1. Nevertheless, the gain in computation time might warrant the slightly lower localization rate. Localization against data set C uses, on average, about 220 image comparisons, compared to the full 943 required in both cases presented above. Of course, as the environment is traversed multiple times, one would expect that the cost for localization in the map would stay constant, whereas it would increase linearly for localization without the map.

The parameter N_{min} is of course important. At a too low value, the localization result will be according to Experiment 1 with the associated performance. At higher values, the localization rate will drop. Choosing a value above the number of correspondences necessary for the epipolar constraint is one way of ensuring that not too many images will be compared; here, we have chosen a value of $N_{min} = 8$ for all data sets.

VII. CONCLUSION AND FURTHER WORK

In this paper, it has been demonstrated that by selecting an appropriate local feature algorithm, using a suitable threshold and the epipolar constraint, a very high rate of localization using only panoramic images can be achieved — even when the images were acquired in different seasons, under different weather conditions. Specifically, the good choices are:

- High-resolution images — necessary to detect features at a detail level that changes little over the seasons.
- U-SURF — very good performance in detecting and matching features in high-resolution, panoramic images.
- Epipolar constraint and RANSAC — improves matching performance at little extra cost.

Further, it has been shown that the incremental spectral clustering algorithm can successfully create topological maps that can be used to perform localization more efficiently (albeit with a slightly smaller success rate).

In this paper, only the case of global localization has been considered, using images from one of the data sets to build the map. There might be advantages in incorporating information from multiple data sets into a single map, to obtain even higher localization rates. The map in itself is also of interest, because it gives a way to divide even outdoor environments into natural, distinctive sections. It might be

of interest to try to automatically label the nodes, based on image content, or to perform path-planning tasks in the map. To this end, it will most likely be necessary to make sure that all false links are eliminated; incorporating odometry information into the affinity matrix can surely help.

VIII. ACKNOWLEDGMENTS

Many thanks to Henrik Andreasson, Martin Magnusson and Martin Persson for their help with the data acquisition.

REFERENCES

- [1] H. Andreasson, T. Duckett, and A. Lilienthal. Mini-SLAM: Minimalistic visual SLAM in large-scale environments based on a new interpretation of image similarity. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 4096–4101, Rome, Italy, 2007.
- [2] H. Andreasson, A. Treptow, and T. Duckett. Localization for mobile robots using panoramic vision, local features and particle filter. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 3348–3353, Barcelona, Spain, 2005.
- [3] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. In *Proc. Ninth European Conf. on Computer Vision*, Graz, Austria, 2006.
- [4] O. Booi, B. Terwijn, Z. Zivkovic, and B. Krse. Navigation using an appearance based topological map. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 3927–3932, Rome, Italy, 2007.
- [5] O. Booi, Z. Zivkovic, and B. Kröse. From sensors to rooms. In *Proc. IROS Workshop From Sensors to Human Spatial Concepts*, pages 53–58, Beijing, China, 2006.
- [6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.
- [7] T. Goedeme, M. Nuttin, T. Tuytelaars, and L. Van Gool. Omnidirectional vision-based topological navigation. *Int. Journal on Computer Vision and Int. Journal on Robotics Research; joint special issue of IJCV and IJRR on vision and robotics*, 74(3):219–236, 2007.
- [8] X. He, R. S. Zemel, and V. Mnih. Topological map learning from outdoor image sequences. *Journal of Field Robotics*, 23:1091–1104, 2006.
- [9] J. Kosecka, F. Li, and X. Yang. Global localization and relative positioning based on scale-invariant keypoints. *Robotics and Autonomous Systems*, 52(1):27–38, 2005.
- [10] B. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119:191–233, 2000.
- [11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2:91–110, 2004.
- [12] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856, 2001.
- [13] 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.
- [14] A. Tapus and R. Siegwart. Incremental robot mapping with fingerprints of places. In *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, pages 2929–2934, Edmonton, Canada, 2005.
- [15] C. Valgren. *Topological Mapping and Localization Using Omnidirectional Vision*. Licentiate thesis, Örebro University, 2007.
- [16] C. Valgren, T. Duckett, and A. Lilienthal. Incremental spectral clustering and its application to topological mapping. In *Proc. IEEE Int. Conf. on Robotics and Automation*, pages 4283–4288, 2007.
- [17] C. Valgren and A. Lilienthal. SIFT, SURF and seasons: Long-term outdoor localization using local features. In *Proc. 3rd European Conf. on Mobile Robotics*, 2007. To appear.
- [18] D. Verma and M. Meila. A comparison of spectral clustering algorithms. Technical Report UW-CSE-03-05-01, University of Washington, 2003.
- [19] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Proc. 18th Annual Conf. on Neural Information Processing Systems*, pages 1601–1608, Vancouver, Canada, 2004.
- [20] W. Zhang and J. Kosecka. Localization based on building recognition. In *Workshop on Applications for Visually Impaired, IEEE Int. Conf. on Computer Vision and Pattern Recognition*, volume 3, pages 21–21, 2005.