Markerless Computer Vision Based Localization using Automatically Generated Topological Maps

Toon Goedemé[1], Marnix Nuttin[2], Tinne Tuytelaars[1] and Luc Van Gool[1,3]
{toon.goedeme, tinne.tuytelaars, luc.vangool}@esat.kuleuven.ac.be, marnix.nuttin@mech.kuleuven.ac.be

[1]PSI-VISICS
K. U. Leuven
Belgium

[2]PMA
K. U. Leuven
Belgium

[3]BIWI
E. T. H. Zürich
Switzerland

1. ABSTRACT

This work was motivated by the goal of building a navigation system that could guide people or robots around in large complex urban environments, even in situations in which Global Positioning Systems (GPS) cannot provide navigational information. Such environments include indoor and crowded city areas where there is no line of sight to the GPS satellites. Because installing active badges or beacon systems involves substantial effort and expense, we have developed a system which navigates solely based on naturally occurring landmarks. As sensory input, we only use a panoramic camera system which provides omnidirectional images of the environment. During the training stage, the system is led around in the environment while recording images at regular time intervals. Offline, these images are automatically archived in a world model. Unlike traditional approaches we don't build an Euclidean metrical map. The used world model is a graph reflecting the topological structure of the environment: e.g. for indoor environments rooms are nodes and corridors are edges of the graph. Image comparison is done using both global color measures and matching of specially developed local features. These measures are designed to be robust, respectively invariant, to image distortions caused by viewpoint changes, illumination changes and occlusions. This leads to a system that can recognize a certain place even if its location is not exactly the same as the location from where the reference image was taken, even if the illumination is substantially different, and even if there are large occluded parts. Using this world model, localization can be done by comparing a new query image, taken at the current position of the mobile system, with the images in the model. A Bayesian framework makes it possible to track the system's position in quasi real time. When the present location is known, a path to a target location can be carried out easily using the topological map.

2. INTRODUCTION AND RELATED WORK

Nowadays, the number of applications using location information is growing exponentially. As localization system, the GPS system's popularity is growing rapidly. However, there are reasons why we did not choose working with GPS, as explained further.

In this paper we describe an alternative localization system with unique specifications. It is applicable in both in- and outdoor urban environments yielding a very high accuracy (typically less than one meter). It also shows to be very easy to train to a new environment. To be able to use it with minimal effort in new environments, our navigation solution does not need to alter the environment. Setting up beacons, markers, badges etc. is time-consuming, expensive and in some cases unaesthetic or even impossible.

We developed a localization method which offers an alternative to the very popular Global Positioning System, or GPS. GPS consists of a constellation of 24 satellites whose transmitted signal can be used to find the location of a mobile system by trilateration. GPS has lots of advantages. It is free for using, invisible, markerless, fast, and can be integrated in a very small device.

The accuracy of GPS is also quite good. Given that the signals come from about 20000 km away of the earth surface, the acquired accuracy of about 15 m can be called astonishingly precise. Nevertheless, a number of applications — like ours — exist that need higher accuracy. That is why methods like *DGPS* [8] and *indoor GPS* [29] are developed. These methods yield an accuracy up to a few cm, but the price that has to be paid is installing extra base stations, what we try to avoid.

For our localization system, we choose not to use GPS as the main sensor. One reason is that GPS only works if there is clear sight to at least four satellites. In indoor situations and in narrow city center streets this is certainly not always the case. Alternatives to GPS have been proposed, including RF devices [2] and systems that use signals of the mobile phone network [30]. But, these methods also need artificial changes in the environment, don't work inside buildings, nor yield an accuracy which is good enough.

A solution is found — as in many cases — in imitating nature. People as well as many animals localize themselves in a known environment mainly by using their eyes. Visual information is especially suited to recognize a place or location. That is why we propose a localization method based on computer vision. The advantages of such a system are numerous: the hardware needed is becoming cheap with off-the-shelve components, it is easy to train in new environments, the installation is simple, ... But, imitating the visual skills of humans is not an easy task. People are surprisingly good at coping with illumination changes, viewpoint differences and occlusion.

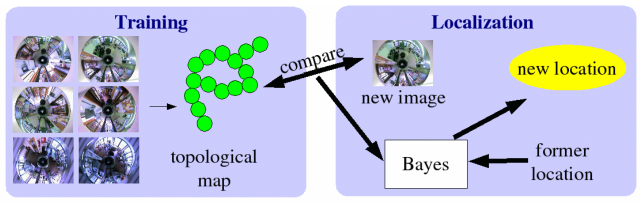

Figure 1: Overview of the localization system

Figure 1 offers an overview of the described system. Our, and most other attempts [11] to build a vision-based localization system acquire a collection of images of the environment during a training stage. Later, a new image taken from an unknown position is compared with the images in the database. The most resembling database image is then considered to give the location of the system.

In our implementation, we opted for an omnidirectional camera system realized by a color camera looking at a hyperbolic mirror. This system, depicted in the top left of fig. 2, produces images with a very wide viewing angle (180°), which is suited for location observat ion. Using these wide-angle images together with viewpoint invariant matching techniques, it is possible to train an environment with substantially fewer images compared to other methods [11]. An example of a recorded image is shown in the top right of fig.2.

A main issue in vision based navigation is how to represent the world model. Many authors propose 2D or even 3D metrical maps (e.g. Davison [3], Fraundorfer [5]). But these methods have a few disadvantages. The detection and recognition is mostly very computationally expensive and memory consuming. The major problem with it is the cumulative error built-up. A more robust alternative is the use of topological representations of the world. In every human-made environment, a certain topological ordering can be observed. Cities can be represented as a graph of streets connecting each other at crossroads. Indoor, a similar graph can be found where the nodes are the rooms that are connected by doorways and corridors. Localization methods that use predefined topological maps include the methods of Ulrich and Nourbakhsh [27] and the method of Tapus, Heinzer and Siegwart [24] propose. It is clear that it is a great advantage if the system itself can construct a topological environment representation using a series of training images. One mathematically sound approach for automatic topological map building is formulated, with very limited experimental validation, by Vale [28].

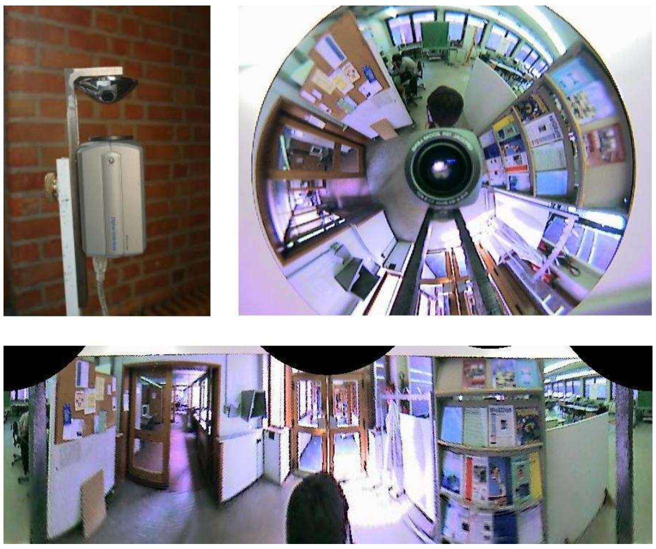

Figure 2: omnidirectional camerea system (top left), example omnidirectional image (top right), transformed image (bottom.)

To be able to construct such a topology and to do localization therein, a method must be found to recognize a certain position in the environment while solely using vision. Global methods exist who look at the entire image, other techniques work at a local scale, extracting and recognizing local features. In our approach, we will combine these two approaches.

Straightforward global methods as histogram-based matching, used by Ulrich and Nourbakhsh [27] and correlation-based template matching, tested by Matsumoto et al. [14] seem not always to work. An other popular technique is the use of a eigenspace decomposition of the training images. This gives the advantage that the image database is very compact, because only the most interesting dimensions of the eigenspace are retained (PCA). Recognition can be done easily by projecting the new image on the eigenspace. Jogan and Leonardis [10] experimented with this technique. However, these methods proved not useful in general situations because they are not robust enough against occlusions (partial coverages of the field of view by an other object) and changes in the illumination of the scene. That is why both Jogan et al. [9] and Paletta et al. [18] developed a PCA-based image comparison that is robust against partial occlusions. Recently, Bischof et al. [1] adapted their

PCA-based image comparison method for use under varying illumination.

A classic approach to recognize images using local features is adding artificial landmarks to the scene. To be easy to detect and recognize these features have special photometric properties (e.g. colored patterns [17] or LEDs [7]) or special geometries (e.g. squares [23], circles [22] or even 2D barcodes [19]). As said before, the use of these special landmarks is in some applications perfectly possible, but in the general case it is not ideal. In our solution, we use only natural local features that are already present in the environment.

The outline of the remainder of this paper is as follows. Firstly, in section 3, the local invariant features we developed to use as natural landmarks are explained. Section 4 describes the two-stage method (global/local) to compare two images. How the topological map building algorithm works, is detailed in section 5. Section 6 introduces our Bayesian localization approach. We end with a description of real-world experiments in section 7 and a conclusion in section 8.

3. LOCAL INVARIANT FEATURES

The technique of local invariant features is originally developed to do *wide baseline matching*, looking for corresponding points between two images that are taken from the same scene, but differ relatively much in viewpoint. Many methods exist to do local feature matching. Both Tuytelaars *et al.* [25, 26] and Lowe *et al.* [13] developed a method for local feature matching which they also used for navigation applications. The main problem with these methods is that the extraction and matching of local features is very time consuming. For example, to search for matches between two 640×480 color images lasts typically half a minute on a recent PC using the technique of Tuytelaars *et al.* For real-time navigation, this is intolerably slow.

That is why in former work [6], we developed a new and fast method for wide baseline matching with local features. Fast execution is achieved by imposing natural constraints on camera pose and scene. A short description of a version for omnidirectional images follows.

3.1 Constraints

As mentioned before, our technique has been developed for use on a mobile system, for instance a mobile robot. Imagine a robot with a fixed omnidirectional camera system mounted on it, which is moving on a horizontal plane. Hence, the camera may (1) translate in the horizontal plane and (2) rotate around a vertical axis, as shown in fig. 3. We also assume that the camera system is oriented vertically.

Furthermore, we assume that the scene contains several vertical elements. This is certainly the case for human-made environments (walls, doors, windows, furniture, etc.), but also natural scenes tend to contain several such structures (e.g. trees). These structures don’t have to be planar, so e.g. cylindrical elements like pillars do comply just the same.

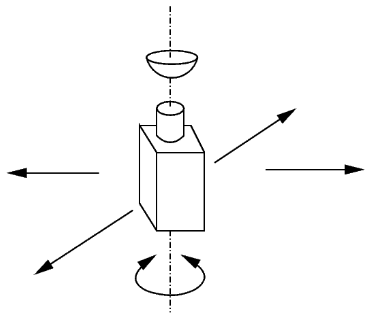

Figure 3: Camera system movement constraints

3.2 Image transformation

As shown in the example of fig. 2, the camera system produces a circular image which is kind of a birds-eye view of the environment of the mobile system. To be able to use the local features described here, a transformation of this image must be performed. We choose for a cilindrical projection, see in the bottom of fig. 2. Doing so, and obeying the constraints on camera position and scene, a vertical line in the world always projects to a scaled vertical line in the image. This enables simple and fast feature extraction.

3.3 Feature extraction and matching

Fig. 4 gives an overview of the algorithm for feature extraction and matching. In our target application, like in many other wide baseline applications, a new image must be matched with one or more prerecorded reference image(s). That is why we split up the algorithm in an off-line part and an on-line part, depicted respectively as the top and bottom row in the figure. While the off-line part – dealing with preparation of the site model – is not time-critical, we try to achieve high-speed execution of the on-line part, where the localization takes place.

In every image, we first extract invariant vertical column segments, the local image features we will describe in section 3.3.1. For each column segment, a descriptor vector is computed based on color and intensity information. Because of repetitive elements in the image, the column segments need to be clustered. To speed up the execution, the data of the reference image(s) is stored in a Kd-tree, in support of fast matching.

3.3.1 Vertical column segment extraction

In this section we describe the affine invariant column segments that we use as features in wide baseline matching. As a matter of fact, every step in their extraction is invariant to the allowable changes in viewpoint and illumination, where the latter amounts to a scaling and offset. The extraction process is illustrated in fig. 5, a part of a transformed omni-

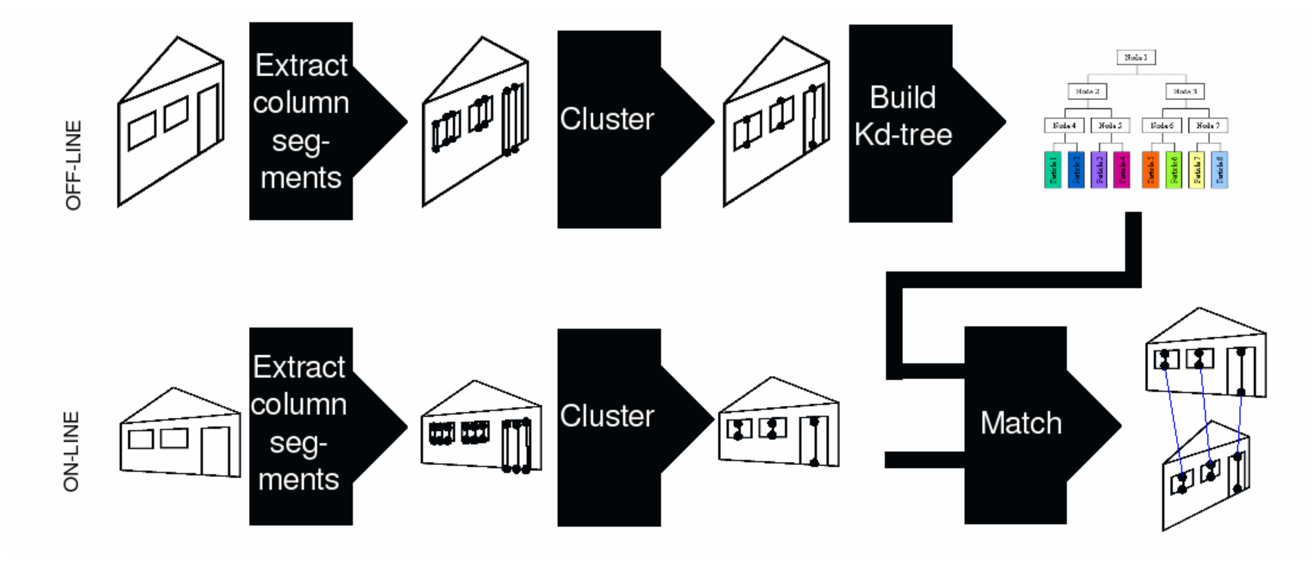

Figure 4: Overview of the column segment extraction and matching procedure

directional image of a door. From the intensity image (B) the gradient magnitude image (C) is computed using the Sobel operator. For every pixel column in the image we look for points having a local maximum gradient value (D). Every consecutive pair of gradient maxima on a column demarcates a new vertical column segment (E). A first selection on the column segments is done comparing the average gradient value between the end points and the gradient value of the end points. If the column segment contains too much noise, this selection criterion eliminates the column segment. To increase the discriminative power of the descriptor vector, the column segment is expanded on both sides with a constant fraction of the segment length (in our experiments 0.2). Fig. (F) shows this.

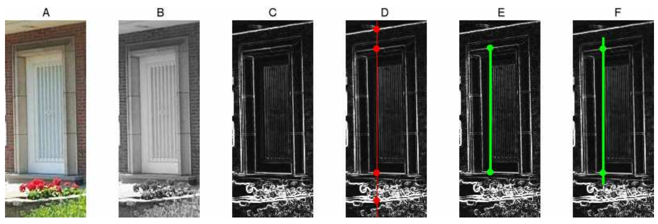

Figure 5: Column extraction algorithm

It is important to see that we do not use edges as features, like in the work of Schmid and Zisserman [21]. The vertical column segment features we extract are just vertical columns of pixels between two points with large gradient value. Because of the simplicity of it, this column segment extraction procedure can be implemented very efficiently.

3.3.2 Descriptor vector calculation

In order to characterize the different invariant column segments, a total of 10 descriptors are used, which combine color and intensity properties:

- Three *color invariants*: To include color information in the descriptor vector, we compute the following color invariants, based on Mindru's 'generalized color moments' [15], calculated over the entire column segment: C_{RB}, C_{RG} and C_{GB}, with

$$C_{PQ}=\frac{\int PQ\,dx\int}{\int P\,dx\int Q\,dx}dx$$

where R, G and B represent the red, green and blue channels, centralized around their means.

- Seven *intensity invariants*. To characterize the intensity profile along the column segment, good features could be obtained through the Karhunen-Loève transform. But because all the data is not known beforehand this is not practical. We have used the *discrete cosine transform* (DCT) instead, as it offers a good alternative in such cases and because it is fast to compute and yields real values. We compute 7 DCT coefficients:

$$F(u)=\sqrt{\frac{2}{N}}\sum_{i=0}^{N-1}\Lambda(i)cos\left[\frac{\pi u}{2N}(2i+1)\ f(i)\right]$$

Where u=1...7, N is the segment length, $f(i)$ is the normalized intensity profile along the column segment, and $1/\sqrt{2}$ if i=0, else $\Lambda(i)$=1. Because of the normalization, the coefficient with u=0 is zero, so it cannot be used. The DCT computations in our algorithm are executed fast using the 8-point 1D method of Loeffler *et al.* [12].

Every step in the extraction of the vertical lines and their descriptors is invariant to changes in viewpoint, if these satisfy the camera motion constraint. In contrast to the invariant regions that are typically used, these features also *perfectly* withstand the effect of perspective projection. Invariance to scene illumination is also guaranteed if it can be modeled by modifying each color band by a scaling and an offset.

For matching and clustering vertical column segments, a distance measure for the description vectors must be defined. In our algorithm, we use the Mahalanobis distance r, with $r^2=(\boldsymbol{X}-\mu)^T\boldsymbol{C_X}^{-1}(\boldsymbol{X}-\mu)$

In this formula $\boldsymbol{\mu}$ and $\mathbf{C_X}$ are the mean and intra-group correlation matrix of $\boldsymbol{X}$, which contains the descriptor vectors. We assume the components of the descriptor vector to be uncorrelated, which leads to a diagonal matrix $\mathbf{C_X}$.

3.3.3 Intra-image clustering

In many cases there are horizontally constant elements in the scene. This leads to contiguous series of similar vertical column segments. To avoid matching over and over again very similar column segments during the on-line navigation stage, on both reference image and query image clustering of the column segments is performed first. As clustering measure we use the Mahalanobis distance of the descriptor vectors, together with the horizontal distance between the line segments.

Each cluster is represented by a prototype column segment, having the average descriptor vector and being centered in the middle of the cluster.

3.3.4 Matching

The goal of the one-column segment matching is to quickly find corresponding vertical column segments (or cluster prototypes) between two images. Here, we define 'corresponding' as having a Mahalanobis distance between the descriptor vectors smaller than a fixed threshold. To speed up the matching, a Kd-tree of the reference image data is built. We used the on-line available package ANN (Approximate Nearest Neighbor) by Mount *et al.*, [16], for this.

Figure 6 gives the matches found between two omnidirectional images. 67 matches were found, in a total processing time of 264 ms on a 2 GHz PC.

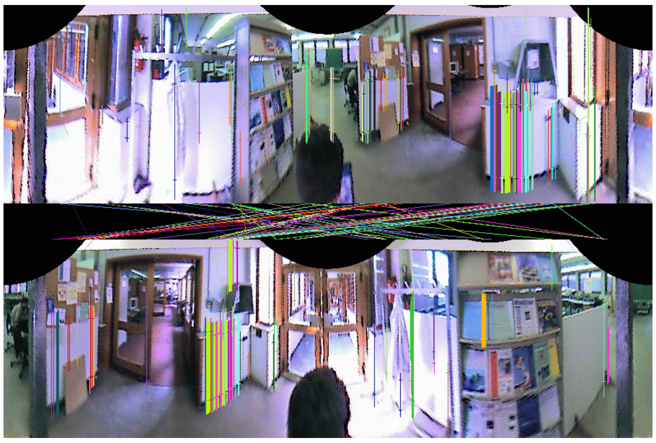

Figure 6: Matches found between two images.

4. COMPARING TWO IMAGES

As explained before, both for the automatic construction of the topological map and localization at a new position, a comparison method for two images is needed. In the map-building phase it concerns two images of the training set, in the localization phase the new image (query image) has to be compared with the images in the database.

Our approach consists of two levels, a global and a local comparison of the images. We first compare two images with a coarse but fast global technique. After that, a relatively slower comparison with more precision based on local features only has to be carried out on the survivors of the first stage. The main goal of this section is giving for each arbitrary pair of images a certain similarity measure, which tells how visually similar the two images are. The inverse of this similarity, the dissimilarity, is then a measure for the physical distance between the positions where the two images are taken.

4.1 Global color dissimilarity

To achieve fast a global image dissimilarity measure between two images, we compute the same moments we used for the column segment color descriptor over the entire image. These moments are invariant to illumination changes, i.e. offset and scaling in each color band. The Euler distance between two sets of these image color descriptors gives a visual dissimilarity measure between two images.

4.2 Dissimilarity measure based on matches

With the former dissimilarity measure, we can clearly see the difference of images taken in different rooms. Because images taken in the same room but at different places have approximately the same color scheme, a second dissimilarity measure based on local features is used.

First, we search for feature matches between the two images. The dissimilarity measure looked for must be inverse proportional to the *number of matches*, relative to the average number of features found in the images. Hence the first two factors in the equation for D_m. But, also the difference in relative configuration of the matches must be taken in account. Therefore, we first compute a global angular alignment of the images by computing the average angle difference of the matches. Figure 7 shows this. The dissimilarity measure is now also made proportional to the average angle difference of the features after this global alignment.

$$D_m = \frac{1}{N}\frac{n_1+n_2}{2}\frac{\sum|\Delta_i|}{N}$$

where N corresponds to the number of matches found, n_i the number of extracted features in image i, Δ_i the angle difference for one match after global alignment.

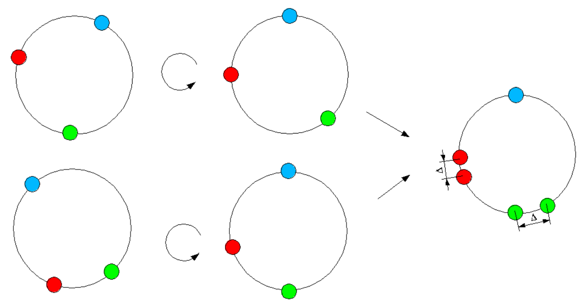

Figure 7: Measuring constellation difference. Form left to right: original images, after global alignment, angle differences measured.

4.3 Combined dissimilarity measure

We observed that the dissimilarity measure based on color is fast but not precise between close viewpoints, while the dissimilarity measure based on matching is less fast and only works for viewpoints, where there are enough matches. Therefor we combine these two measures: only those pairs of images who have a color dissimilarity under a predefined threshold are candidates for computing a matching dissimilarity. If we scale the latter, this creates one dissimilarity measure plotted in fig. 8.

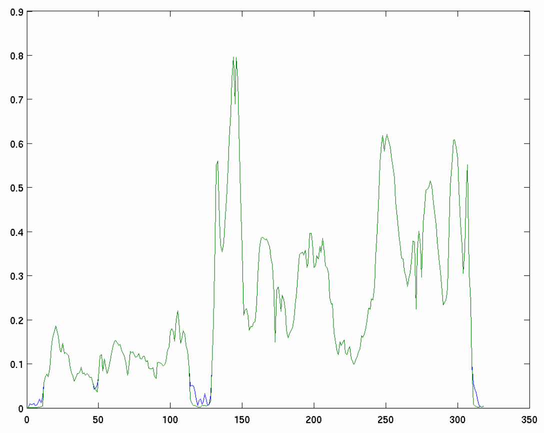

Figure 8: Cobined dissimilarity measure (horizontal axis: image number). Green: global color dissimilarity. Blue: dissimilarity based on matches.

5. TOPOLOGICAL WORLD MODEL

One of the qualities of the here described localization method is that it is able to construct automatically a topological world representation out of a sequence of training images. During a training tour through the entire environment, omnidirectional images are taken at regular time intervals. The order of the training images is known.

Figure 9 gives a small example to explain the algorithm used. An S-shaped path was traveled, yielding images depicted as small yellow rectangles in the figure (a). We observe that the spatial density of the training images is not homogeneous, because the exploring speed is not constant. Therefore, *place clustering* is performed. The dissimilarity measure of section 4.3 is computed between consecutive images, and *places* with constant size are formed (green ellipses in (b)). For each place, a prototype image is formed with mean properties. A next phase in the algorithm performs an exhaustive comparison of all non-consecutive place prototypes to find locations that are visited more than once during the training session. The place prototype pairs with a dissimilarity under a predefined threshold get an extra connection between them (c). This yields the topological map depicted in figure (d).

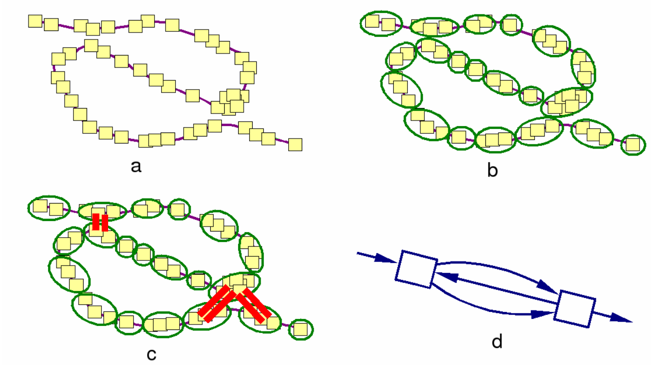

Figure 9: Algorithm for automatic topological map building

6. LOCALIZATION

When the system has learned a topological map of an environment, this map can be used for navigational tasks. The task we look at in this work is *localization*. For each arbitrary new position in the known environment, the system can find out *where* it is. The output of such a localization algorithm is a *location*, which is — opposed to other methods like GPS — not expressed as a metric coordinate. In our topological approach a location is expressed as a reference to the training image that is the closest to the current position.

The training set doesn't need to cover every imaginable position in the environment, in contrast to [11]. A relatively sparse coverage is sufficient to localize every possible position. That is because the image comparison method we developed is based on wide baseline techniques, hence can recognize scene items from substantially different viewpoints.

Actually, two localization modes exist. When starting up the system, there is no a priori information on the location. Every location is equally probable. This is called global localization, alias the kidnapped robot problem. Traditionally, this is known to be a very hard problem in robot localization. In contrary, if there is knowledge about a former localization not too long ago, the locations in the proximity of that former location have a higher probability than others further away. This is called location updating.

We propose a system that is able to cope with both localization modes. A probabilistic approach of this matter is suitable. Instead of making a hard decision about the location, a probability value is given to each location at each time instant. The Bayesian approach we follow is explained in the next section.

6.1 Bayesian Filtering

Define $x \in X$ a place of the topological map. Z is the collection of all omnidirectional images z, so that $z(x)$ corresponds to the training observation at place x. At a certain time instant t, the system acquires a new image z_t. The goal of the localization algorithm is to reveal the place x_t where this image was taken.

We define the *Belief function* $Bel(x,t)$ as the probability of being at place x at time t, given all previous observations. So,

$$Bel(x,t)=P(x_t|z_t,z_{t-1},\ldots,z_0),$$

for all $x \in X$. In the *kidnapped robot* case, there is no knowledge about previous observations hence $Bel(x,t_0)$ is initialized equal for all x.

Using Bayes' rule, we find:

$$Bel(x,t)=\frac{P(z_t \mid x_t, z_{t-1},\ldots,z_0)P(x_t \mid z_{t-1},\ldots,z_0)}{P(z_t \mid z_{t-1},\ldots,z_0)}$$

Because the denominator of this fraction not dependent on x, we replace it by the normalizing constant η. Also, the probabilistic sum rule is used to obtain

$$Bel(x,t)=\eta P(z_t \mid x_t, z_{t-1},\ldots,z_0)\cdot \sum_{x_{t-1}\in X}\left[P(x_t \mid x_{t-1}, z_{t-1},\ldots,z_0)P(x_{t-1} \mid z_{t-1},\ldots,z_0)\right]$$

If we know the current location of the system, then future locations do not depend on past locations. This property is called the *Markov Assumption*: $P(z_t|x_t,z_{t-1},\ldots,z_0)=P(z_t|x_t)$. Using it, we yield:

$$Bel(x,t)=\eta P(z_t \mid x_t)\sum_{x_{t-1}\in X}\left[P(x_t \mid x_{t-1})Bel(x,t-1)\right]$$

This allows us to calculate the belief recursively based on two variables: the *next state density* or *motion model* $P(x_t|x_{t-1})$ and the *sensor model* $P(z_t|x_t)$.

6.2 Motion Model

The motion model $P(x_t|x_{t-1})$ explicits the probability of a transition from one place x_{t-1} to another x_t. It seems logical to assume that a transition in the time instant $(t\text{-}1) \rightarrow t$ between places who are far from each other is less probable than between places close to each other. We model this effect with a Gaussian:

$$P(x_t|x_{t-1})=\frac{1}{\beta_x}e^{-dist(x_{t-1},x_t)/\sigma_x^2}.$$

In this equation, the function $dist(x_1,x_2)$ corresponds to a measurement of the physical distance between the two places. We approximate it as the minimum number of place transitions to go from x_1 to x_2 on the topological map. This number, computed with the Dijkstra algorithm [4], is a good indication for the real distance between x_1 and x_2. Indeed, the place clustering is so that every place has the same size, and the (visual) distances between connected places is therefor constant. β_x is a normalization constant, and σ_x^2 is the variance of the distances, measured on the map data. Once the topological map is known, the complete motion model can be computed off-line for usage during localization.

6.3 Sensor Model

The entity $P(z_t|x_t)$, called the sensor model, is the probability of acquiring a certain observation z_t if the location x_t is known. This is related to the visual dissimilarity of that observation z_t and the training observation $z(x_t)$ at location x_t. The probability of acquiring an image at a certain place that differs much from the training image taken at that place has a low probability. That is why we model this sensor model also by a Gaussian:

$$P(z_t|x_t)=\frac{1}{\beta_z}e^{-diss(z_t,z(x_t))/\sigma_z}.$$

This time, the function $diss(z_1,z_2)$ refers to the visual dissimilarity explained in section 4. σ_z is the standard deviation of it, measured on the data and β_z is a normalization constant.

Unlike the motion model, the sensor model cannot be computed beforehand. It depends on the new incoming query image data. Every location update step the visual dissimilarities of the query image with all the database images must be computed. This validates our efforts to make the computation of the visual dissimilarity measure as fast as possible.

7. EXPERIMENTS

Figure 10 shows some of the training images out of a sequence of 498, taken during a run through our robot lab. It also shows the true trajectory of that training run, shown on a metrical map. The automatically generated topological map is shown in figure 11. It is clear that this map reflects the topology of the training run.

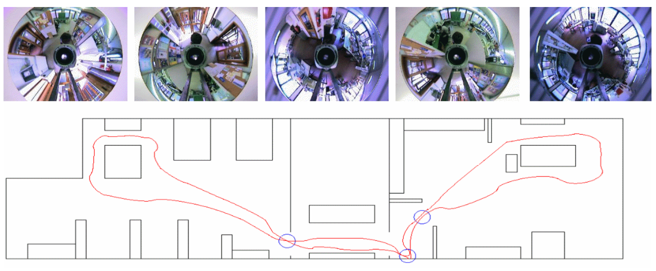

Figure 10: Some of the training images and the training trajectory.

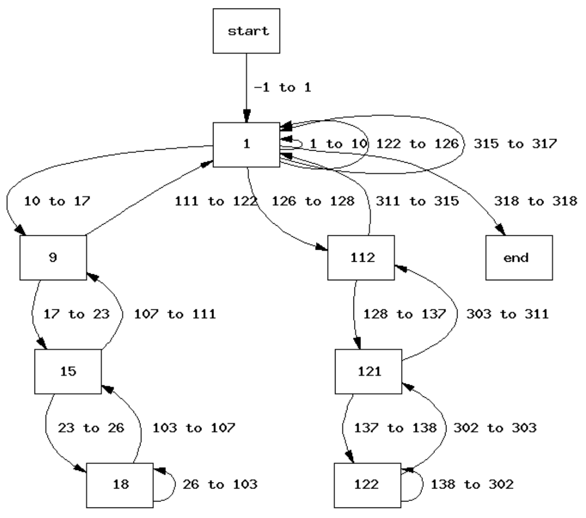

Figure 11: The extracted topological map.

To test the localization performance, 100 extra images were taken at the robot lab. Some of them on the learned path, some further away from it. We found that, for 72% of these query images, the maximum of the belief function after the first iteration is already pointing to the right position. In 95% the believed location is right after the third iteration. This shows the power of the Bayesian updating principle, wherein the new observation and former location belief information is merged. The average total belief function update time was 1245 ms on a 2 GHz PC.

Figure 12 shows the results of a small localization experiment. The robot was placed sequentially at places 2, 3 and 4. In each of these positions a new query image was taken and the uniformly initiated belief function was updated, as shown in the colored graph. It is clear that indeed the maximum of the belief function points to the location of the system. When more knowledge of former localizations is collected, the belief function is sharper, i.e. more confident of the location.

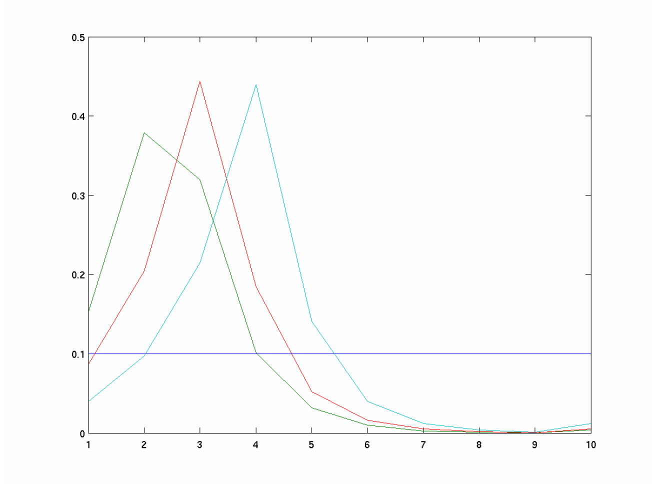

Figure 12: Example of continually updated belief function.

8. CONCLUSION

In this paper we proposed, described and tested an alternative localization method. A full vision based system is proposed which can offer an alternative to GPS. It is not dependent on satellites, extra base stations, artificial markers or beacons and is easy to train in a new environment. The latter is achieved by using wide baseline matching techniques on wide angle omnidirectional images. Unlike other vision based localization methods, the proposed system is robust to occlusions, illumination changes and severe viewpoint differences. A topological map based Bayesian localization system is able to track the system's position at 0.75 Hz.

9. ACKNOWLEDGEMENTS

This research has been carried out with the financial support of the Inter-University Attraction Poles, Office of the Prime Minister, IUAP-AMS, the Flemish Institute for the Advancement of Science in Industry, IWT, and the Fund for Scientific Research Flanders (FWO-Vlaanderen, Belgium).

10. REFERENCES

[1] H. Bischof, H. Wildenauer, and A. Leonardis, *Illumination insensitive eigenspaces*, In Proc. ICCV01, pages 233-238.IEEE Computer Society, 2001.

[2] N. Bulusu, J. Heidemann and D. Estrin, *GPS-less Low Cost Outdoor Localization for Very Small Devices*, IEEE Personal Communications Magazine, Vol. 7, No. 5, pp. 28-34. October, 2000.

[3] A. Davison, Y. Cid and N. Kita, *Real-Time 3D SLAM with Wide-Angle Vision*, Proc. IFAC Symposium on Intelligent Autonomous Vehicles, 2004.

[4] E. W. Dijkstra, *A note on two problems in connexion with graphs*, Numerische Mathematik, 1: 269-271, 1959.

[5] F. Fraundorfer, *A map for mobile robots consisting of a 3D model with augmented salient image features*, 26th Workshop of the Austrian Association for Pattern Recognition (ÖAGM/AAPR) 2002, pp. 249-256, 2002.

[6] T. Goedemé, T. Tuytelaars, and L. Van Gool, *Fast Wide Baseline Matching with Constrained Camera Position*, To appear: Conference on Computer Vision and Pattern Recognition, Washington DC, 2004.

[7] W. Hoff, *Fusion of data from head-mounted and fixed sensors*, 1st Int. Workshop on Augmented Reality, Nov.1, 1998, San Francisco

[8] J. Hurn, *Differential GPS explained*, Trimble Navigation, 1993.

[9] M. Jogan, A. Leonardis, *Robust localization using panoramic view-based recogniton*, In Proceedings

15th International Conference on Pattern Recognition ICPR'00, pages 136-139, Barcelona, Spain.

[10] M. Jogan, and A. Leonardis, *Panoramic eigenimages for spatial localisation*, In Solina, Franc and Leonardis, Ales, Eds. Proceedings 8th International Conference on Computer Analysis of Images and Patterns(1689), pages 558-567, Ljubljana, 1999.

[11] B. Kröse, J. Porta, A. van Breemen, K. Crucq, M. Nuttin, and E. Demeester, *Lino, the User-Interface Robot*, First European Symposium on Ambient Intelligence (EUSAI 2003), pp. 264-274, Veldhoven, The Netherlands, 2003.

[12] C. Loeffler, A. Ligtenberg and G. Moschytz, *Practical Fast 1-D DCT Algoritmh with 11 Multiplications*, ICASSP, pp. 988-991, 1989.

[13] D. Lowe, *Object Recognition from Local Scale-Invariant Features*, ICCV, pp. 1150-1157, 1999.

[14] Y. Matsumoto, K. Ikeda, M. Inaba, H. Inoue, *Visual Navigation using Omnidirectional View Sequence*, Proceedings of 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'99), pp.317-322, 1999.

[15] F.Mindru, T. Moons, and L. Van Gool, *Recognizing color patters irrespective of viewpoint and illumination*, CVPR vol. 1, pp. 368-373, 1999.

[16] S. Arya, D. Mount, N. Netanyahu, R. Silverman, and A. Wu, *An optimal algorithm for approximate nearest neighbor searching*, J. of the ACM, vol. 45, pp. 891-923, 1998, http://www.cs.umd.edu/~mount/ANN/.

[17] T. Okuma, K. Sakaue, H. Takemura, and N. Yokoya, *Real-time camera parameter estimation from images for a Mixed Reality system*, Proc. ICPR, Barcelona, Spain, 2000.

[18] L. Paletta, S. Frintrop, J. Hertzberg, *Robust localization using context in omnidirectional imaging*, In: 2001 IEEE Intl. Conf. on Robotics and Automation, pp. 2072-2077, Seoul, Korea, 2001.

[19] J. Rekimoto and Y. Ayatsuka, *CyberCode: Designing Augmented Reality Environments with Visual Tags*, Designing Augmented Reality Environments (DARE 2000), 2000.

[20] José Santos-Victor, Raquel Vassallo, Hans-Jorg Schneebeli, *Topological Maps for Visual Navigation*, VisLab-TR 01/99, 1st International Conference on Computer Vision Systems, Las Palmas, Canarias, January 1999.

[21] C. Schmid and A. Zisserman, *Automatic line matching across views*, Proceedings Conference on Computer Vision and Pattern Recognition, pp. 666-671, 1997.

[22] V. Sundareswaran, and R. Behringer, *Visual Servoing-based Augmented Reality*, in Procs. Intl. Workshop on Augmented Reality, San Francisco, Nov 1, 1998.

[23] D. Tan, I. Poupyrev, M. Billinghurst, H. Kato, H. Regenbrecht, and N. Tetsuani, *The Best of Two Worlds: Merging Virtual and Real for Face-to-Face Collaboration*, ACM CSCW 2000: Workshop on Shared Environments to Support Face-to-Face Collaboration. Philadelphia, Pennsylvania, USA, December 2000

[24] A. Tapus, S. Heinzer, and R. Siegwart, *Bayesian Programming for Topological Global Localization with Fingerprints*, In Proceedings of the IEEE International Conference on Robotics and Automation, 2004.

[25] T. Tuytelaars, L. Van Gool, L. D'haene, R. Koch, *Matching Affinely Invariant Regions for Visual Servoing, International Conference on Robotics and Automation*, Detroit, pp. 1601-1606, May 10-15, 1999.

[26] T. Tuytelaars and L. Van Gool, *Wide baseline stereo matching based on local, affinely invariant regions*, Proc. British Machine Vision Conference, Vol. 2, pp. 412-425, Bristol, 11-14 sept, 2000.

[27] I. Ulrich and I. Nourbakhsh, *Appearance-Based Place Recognition for Topological Localisation*, IEEE International Conference on Robotics and Automatisation, San Francisco, CA, April 2000, pp. 1023-1029

[28] A. Vale, M. Isabel Ribeiro, *Environment Mapping as a Topological Representation*, Proceedings of the 11th International Conference on Advanced Robotics - ICAR2003 Universidade de Coimbra, Portugal, June 30 - July 1- 3, 2003.

[29] F. van Diggelen, C. Abraham, *Indoor GPS Technology*, CTIA Wireless-Agenda, Dallas, TX, 2001.

[30] G. Wölfle, R. Hoppe, D. Zimmermann, and F.M. Landstorfer, *Enhanced Localization Technique within Urban and Indoor Environments based on Accurate and Fast Propagation Models*, European Wireless 2002, Firence (Italy), Feb. 2002.