Vision Based Intelligent Wheel Chair Control: the role of vision and inertial sensing in topological navigation

Toon Goedemé[1], Marnix Nuttin[2], Tinne Tuytelaars[1], and Luc Van Gool[1,3]

[1]ESAT - PSI
University of Leuven
Belgium

[2]PMA
University of Leuven
Belgium

[3]BIWI
ETH Zürich
Switzerland

[toon.goedeme, luc.vangool]@esat.kuleuven.ac.be, marnix.nuttin@mech.kuleuven.ac.be

Abstract

This paper describes ongoing research on vision based mobile robot navigation for wheel chairs. After a guided tour through a natural environment while taking images at regular time intervals, natural landmarks are extracted to automatically build a topological map. Later on this map can be used for place recognition and navigation. We use visual servoing on the landmarks to steer the robot. In this paper, we investigate ways to improve the performance by incorporating inertial sensors.

1 Introduction

In this paper we aim at intelligent electronic wheel chair control. This work results from a cooperation between researchers of the departments of computer vision, mechanical engineering and agriculture of several Belgian universities, in the framework of the project AMS-WP4[1]. The common goal is the development of systems making electronic wheel chairs more intelligent.

Next to the items of shared control, precision guidance and intention extraction in the AMS project [1, 2] the research subject of this paper lies in the field of global navigation. We follow a localization approach at a higher cognitive level. We are in the process of developing a system that builds automatically a topological map of a large-scale environment such as a building or a city. Using this map, localization and navigation tasks can be performed.

The approach is mainly developed based on visual sensors. At this moment, we make use of omnidirectional cameras based on hyperbolic mirrors and fisheye lenses. Nevertheless, we are investigating ways to add other sensors, to increase the robustness and performance of the system. Inertial sensors seem to provide the complementary information we need.

The paper is organized as follows. Section 2 explains the vision based navigation application under construction. The combination of this system with inertial sensors is discussed in section 3. A short description of our test-platform follows in section 4. Section 5 gives an overview of the experimental results we have so far. We end with a summary and conclusions in section 6.

2 Vision based navigation

Autonomous robots still often use special markers for vision based localization [3, 4, 5]. This is especially the case in cluttered scenes, where obstacles are difficult to pick up from laser scanner data, a rather expensive sensor compared to vision. Also in wide open fields when objects are too far to be in the range of the laser scanner visual landmarks offer a way out. In this project we use a novel kind of *natural landmarks* that are available in most scenes. Hence, no special markers are required, as these landmarks can be extracted from the scene itself. Moreover, the extraction of these landmarks is automatic and robust against changes in viewpoint and illumination.

The aim is to build a topological model of the environment, purely based on information from images taken during a guided tour through the environment. Unlike the often used metrical world models such as maps and 3D models, it concerns a graph-like representation of the topological structure of the environment. Streets, for instance, become edges that connect the crossroads and squares in a town. An indoor environment can be modeled at the level of corridors connecting rooms. As a consequence, the robot can navigate through the environment at a cognitive level.

The robot will then be able to repeat the learned trajectories by comparing its current position against those from where the example images have been taken. The robot can find its position even when it is off course and under different illumination conditions,

[1] http://www.mech.kuleuven.ac.be/pma/project/ams

due to the invariance of the natural landmarks. The robot will automatically adapt its position to match it against the most similar example image, based on visual servoing techniques.

An important increase in performance and robustness can be achieved when combining this visual system with inertial sensors. In a way these sensors can have the same functionality as the visual system. We can implement these two systems redundantly to each other. When one system fails the continuation is guaranteed in that way. On the other hand inertial sensors can work at a higher accuracy (for a short time period) and much higher frequency than the visual system. So they can be used to complement the visual system. In section 3 this issue is discussed further.

2.1 Extraction of natural landmarks

We work with local invariant image patches as natural landmarks. These features are developed in the context of wide baseline matching, the process of finding correspondences between two images of the same scene that are taken from viewpoints that lie rather far from each other. But they are useful in the field of robot navigation as well.

There exist several methods to do wide baseline matching, for instance [6, 7, 8, 9, 10, 11, 12]. Tell [11] gives a quite complete overview. At a first stage most of these methods extract *interest points*, such as harris corners or local intensity extrema in the image. Secondly they select a neighborhood around an interest point or between interest points in a viewpoint invariant way and compute feature values, or descriptors, which describe the content of that image patch in a compact and invariant way. For each interest point, these values are stored in a feature vector. To do wide baseline matching it suffices to extract these neighborhoods in each image and search for neighborhoods with identical (or similar) descriptors.

We chose the technique developed by Tuytelaars and Van Gool [12, 13] to use in this project. The power of it lies in the fact that every extraction step of the local image patches is invariant to affine changes in viewpoint and illumination. They define for a region around an interest point a function A(s). The parameter s controls the size and shape of the region. The function A(s) is selected in such a manner that the shape parameters corresponding to its local extrema define a region outline in an affine invariant way. Hence, by selecting regions where local extrema of A are found, the same support regions for computing descriptors can be used irrespective of the viewpoint. Tuytelaars *et al.* suggest several ways of defining the region. One is by defining a parallelogram where two sides are controlled by the edges connecting in a harris corner. Another way is growing an elliptical region around a local intensity extremum.

The descriptor computation we use is based on colour moments up to the first order [14]. The moments of the different colour bands are combined to achieve invariance to scaling and offset changes. However, this is only possible if the descriptors are computed for the same region of the viewed scene in both images. The previously explained invariant region extraction guarantees this.

Finding matches between two images can be achieved by trying to find for each region in one image a region with a similar descriptor in the other image. The crux of the matter is that the extraction of the regions in each image can be done completely independently. This allows to build a database of feature vectors, which can be searched very fast using indexing [15].

We plan to use these natural landmarks in two ways. Firstly, they make it possible to recognize places, a feature we will use intensively while building a topological map of the environment. A second way in which we use these landmark matches is during the actual navigation, to steer the robot from one viewpoint to another.

Figure 1 shows the extracted regions in two images, taken by a omnidirectional camera on a robot. The lines connecting regions show the matches found between these images. The number of matches between two robot images give a good similarity measure for the pose of the robot. To be independent of the complexity of the scene this measure must be divided by the average number of regions found in one image. The similarity s_{ij} between two images i and j can thus be calculated as:

$$s_{ij} = \frac{\#(r_i \equiv r_j)}{\frac{\#r_i + \#r_j}{2}}, \qquad (1)$$

where r_i is a region in image i, and the $\equiv$ sign means a match. Because maximally only 20% of the regions in one image can be matched, a similarity of 0.2 is considered as very resembling.

2.2 Topological representation

As explained in the introduction, we want to build a world model that reflects the topological structure of the world the robot is wandering through. We target at both indoor and outdoor environments.

In human made environments, such as cities and buildings, we often observe a clear structure or ordering in space. Cities, for instance, are nothing more than a maze of streets connecting each other. This makes it easier for human beings to navigate around in that environment. The clearer the topological structure, the easier it is for us to remember trajectories, to reason about navigational issues, and to communicate trajectorial instructions to others.

Such a topological structure can be shown in a graph. Figure 2 for instance shows a topological map

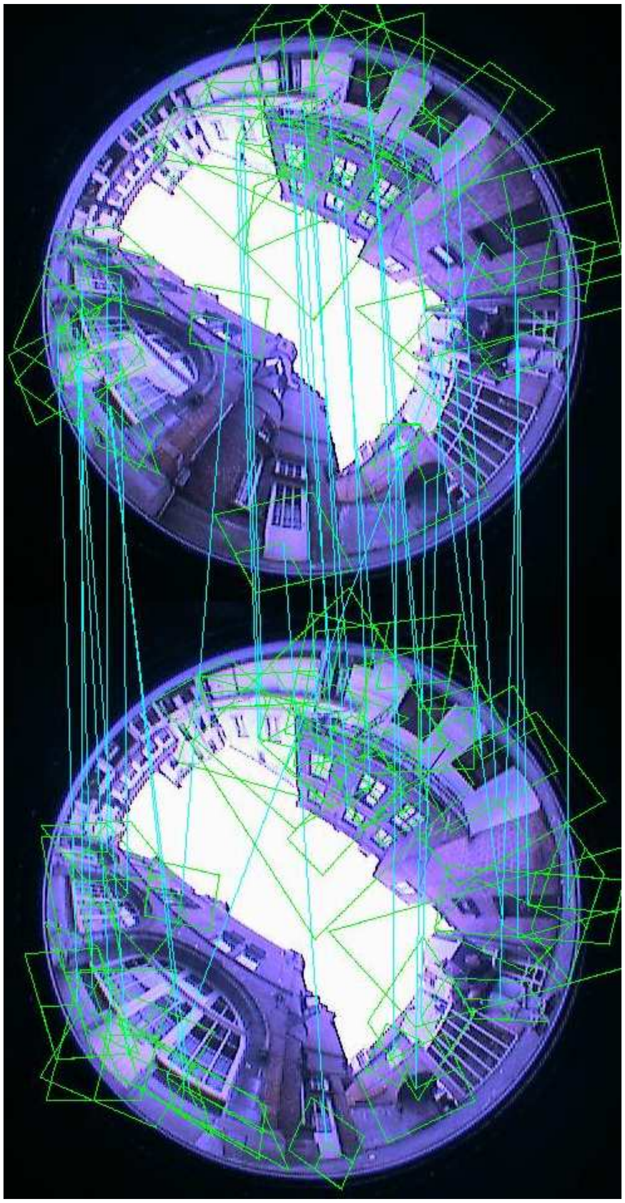

Figure 1: Two omnidirectional images, with affine invariant regions (rectangles) and matches (lines). The images were taken with a distance in between of approximately 2 m. In the upper image 171 affine invariant regions were extracted, in the bottom one 179. There were 35 matches found between this two sets of regions, shown by the lines. This yields a similarity measure of 0.10.
As an example how to read this image, look at the white door in the bottom of each image. A rectangle around it can be seen in both images, which shows that for each image a region around that door is found. The vertical line connecting the center points of these two rectangles shows that these regions have been matched to each other.

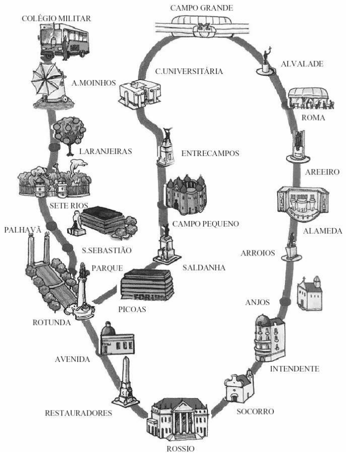

Figure 2: *A topological map of touristic landmarks in Lisbon, Portugal.*

of the touristic landmarks in Lisbon, Portugal. This decomposes the routefinding task in smaller pieces. In order to navigate this environment the only things one needs to know are this map, and how to get from one landmark to another.

2.3 Building the topological graph

It is clear that this topological information in the environment is of great value for applications like robot navigation. In opposition to classical approaches [16, 17], the here described method relies completely on a topological representation of the environment, without the use of metrical maps. Out of images taken at regular time intervals during a guided tour, the system extracts automatically the topological structure. To this end, we cluster the images that belong to one distinct place (like a room). Between these clusters there will be links showing the traversable paths between these places.

We can apply the similarity measure of equation 1 on every pair of images of the test run to look for images that are taken in the same room. The similarity measure is high when there are relatively many matching regions between two images. This can mean that they are taken from a location that is not very different. The aim is now to cluster all the images that belong to one 'place'.

It is possible to construct a graph with images as vertices, connected with links with weight s_{ij}. A simple example is given in figure 3. We can apply graph clustering techniques to cluster together the images that belong to the same place.

Currently, we are still trying to find the optimal clustering algorithm, but a first quick and cheap technique we tested is histogram thresholding. By inspecting the histogram of the link weights of the graph it is possible to automatically set a threshold. When

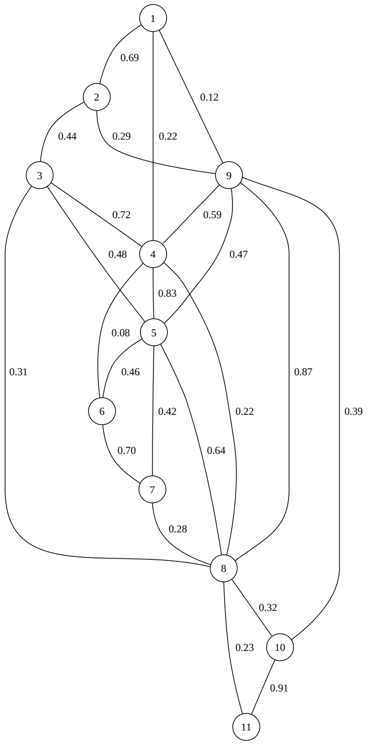

Figure 3: *An example graph of images with similarity links.*

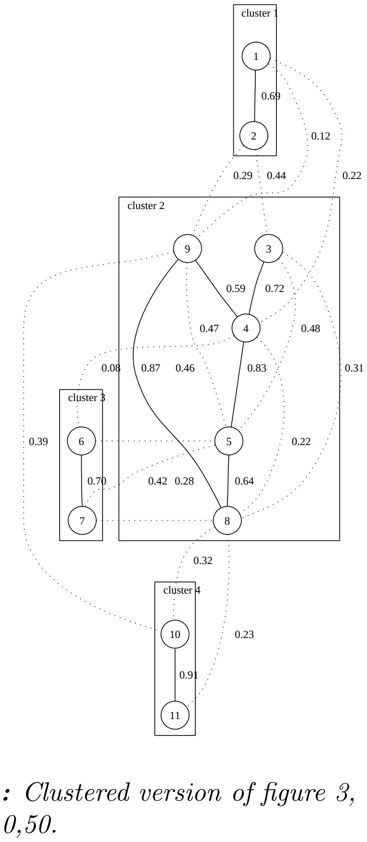

Figure 4: *Clustered version of figure 3, using weight threshold 0,50.*

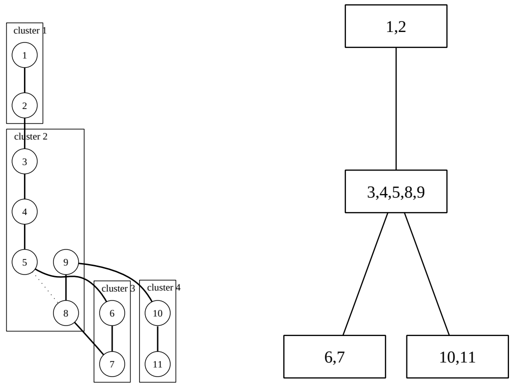

Figure 5: *Derived topological maps for the example.*

links below this threshold are neglected, the graph falls apart in a number of distinct subgraphs that can be seen as clusters. Figure 4 illustrates this. Later steps will aim at improving this raw clustering. Because we know the order of the images during the guided tour, we can link the clusters by traversable paths (bold lines in figure 5, left). This leads to the final topological map, making abstraction of the images, shown in figure 5 on the right hand side.

2.4 Robot localization

During the training tour through the entire environment, the system takes images at a constant rate. For the outdoor case the robot takes images with a distance of 2 meters in between. All the affine regions and their descriptors are computed off line, and put in an indexed database. For an image taken from a new viewpoint the similarity measure of equation 1 can be computed for all the images in the database.

In this way localization is easily done. The place of the robot is equal to the place where the most resembling image is taken. Even the *kidnapped robot* problem is solved. Of course this is only possible if the extracted regions are descriptive enough. In the future we plan to exploit not only the number of matches, but also their geometrical position w.r.t. each other to get a more distinctive similarity measure.

2.5 Steering the robot

As explained above, when the topological map is known and localization can be done, the only remaining necessary skill is point to point navigation, i.e. moving from a certain position to a predefined reference position. With such a technique all the previously recorded travels from one training image to the next can be replayed. In that way we can re-execute every path that the robot traveled through previously. In case there are distinct paths inside a cluster the most similar image pair can build a bridge between these paths (for instance the dashed edge in the left of figure 5). This will turn the topological graph in a maze of connections between the clusters.

We plan to tackle the point to point navigation issue by visual servoing. Because the matches of affine invariant regions provide information about the relative position (up to a scale factor), we can use them to steer the robot from one position to the target position.

For a transition from an image taken from one viewpoint to another we can compute the essential matrix E [13]. From this, the relative rotation and the direction of the relative translation can be determined. It is shown that eight corresponding points between the images are all that is needed to compute the essential matrix using linear techniques. If non-linear techniques are applied, even fewer point correspondences will suffice. Because of the similarity of two consecutive images, the necessary number of wide baseline point matches are guaranteed. After the movement is started in the computed direction, a third image gives information of the progress so far. In that way also the magnitude of the relative translation can be computed.

During the movement new images are acquired, and the rotation and translation to the goal position are updated. It is clear that for every new incoming frame the affine invariant region extraction and matching has to be repeated. Because of the time complexity of this it is far more efficient to track the formerly found regions from one frame to the next. This is done by the algorithm developed by Ferrari *et al.* [18].

In the case of a camera that is mounted fixed on a robot, the problem is in an important manner reduced. We know that a robot is moving mostly in a horizontal plane, while only changing its orientation around a vertical axis. How we can use these facts to reduce the computation time of our application is subject to ongoing research.

2.6 Related work

The technique of *natural landmarks* has inspired several researchers to build a robot navigation system. Se, Lowe and Little for instance use scale-invariant natural landmarks for a *simultaneous localization and mapping* (SLAM) approach [16]. Out of matches between different positions they create a 3D model of the environment. From that model they derive a metrical map. Livatino and Madsen [17] find natural landmarks by cross-correlating image parts with prerecorded scene patches. They compute the metrical position of the robot by doing a triangulation on an optimal triplet of landmark appearances in the image. Davison and Murray [19] use an active stereo head to track natural landmarks, extracted with the operator of Shi and Tomasi [10]. They do metrical localization with a vector-based approach.

None of these methods use natural landmarks that are invariant under affine geometrical deformations. This implies that the space from where a region can be recognized will be significantly smaller compared to when using affine invariant features. Also noticeable is the fact that all these methods make use of a metrical map, while this paper describes an approach based on a topological map.

The advantage of a topological map is the ease for path planning. A path form one position to another can directly be extracted from the graph. Other research recognized this advantage, and developed robot localization and navigation systems based on a topological approach. Ulrich and Nourbakhsh [20] created a method for appearance-based place recognition for topological localization. Their similarity measure, analog to ours in equation 1, is based on histogram matching. A serious drawback of their system is that it does not extract the topological structure itself, this has to be done manually. Franz *et al.* [21] studied topological navigation from a more theoretical point of view. They built a simple robot that builds a topological map. As landmarks they use global features of snapshot images. The topic of the research of Gross *et al.* [22] is the development of a robot driving around in a supermarket. They use a topological map that is closely linked to the metrical reality. They perform Monte Carlo Localization, using global image features of a omnidirectional camera as landmarks.

All these methods use global features for place recognition, which are less robust against e.g. occlusions compared to the local features we use.

3 Combination with inertial sensing

In principle, a navigational system can operate perfectly when relying solely on visual information. But nevertheless adding another independent sensor system can only increase the overall performance. This is also the case by humans. When walking around, we navigate mainly using the information that comes from our eyes, but it is a fact that we use also some kind of odometry (like using the vestibular sensor in the ear, or more or less counting footsteps while walking). The combination of two sensory systems has important merits. On the one hand, when one of the two independent systems fails, the other can take over the functionality. On the other hand, combining the measurements of the two systems reduces the error and increases the performance.

Besides vision, in this project we are planning to add an inertial sensor system, composed by an accelerometer and a gyroscope measuring respectively the change in velocity and the change in rotation. The signals of these sensors are to be fused to yield position information. Such a inertial position system shows a considerable amount of drift for longer periods in time or distance [23]. That is why we shall restrict the use of it to the relative short manoeuvre during the point to point navigation between two successive training images.

Doing this makes the system more robust. There are many reasons why the vision based navigation sometimes drops in performance or even fails completely: a big object like a truck occluding an important piece of the field of view, a place where it is dark because the lights are switched off, a clean textureless white wall as in hospitals,. . . It is clear that we are not always in the case that enough matches can be found between each consecutive pair of images. If during the training phase the movement is recorded with the inertial sensors also, for short distances the navigation can be done using only inertial information.

Moreover, next to the parallel use of vision and inertial sensing, a real combination of the two sensors can be figured out. Inertial sensors have the advantage that they can work on a very high frequency, compared to the slow process of image acquisition and processing. An opportunistic system exploits this. To win time, during the few seconds phase of image acquisition and computing, the robot can be put in motion steered by inertial information. As soon as the data is available from the image, the navigation is updated.

The information of the inertial sensors can be used in two more ways. While region tracking, it provides a very good estimation of the position of a region in the next frame [24]. This will make the tracking faster and more accurate. This is especially useful when the scene contains a repetition of the same item, for instance windows on a building facade. The chance of getting confused decreases when using an inertial estimate.

Last but not least, inertial position measurements make the calculation of the magnitude of the relative translation in visual servoing more accurate. Now the movement between first and intermediate viewpoint can be calculated exactly. Using the essential matrix the target position can be derived with much more accuracy.

4 The test-platform

The test-platform is called Sharioto and is shown in figure 6. Sharioto is a wheel chair equipped with a CAN-bus (called DX-bus by the manufacturer). A laptop emulates many functions of the existing system and is able to redirect velocity commands from the joystick via the laptop to the motors. The parallel port is interfaced to a CAN-bus driver. The joystick and the power module driving the motors are connected via the CAN-bus. A Sony DFW-500VL camera with FireWire (IEEE 1394) connection, in combination with a hyperbolic mirror and a fish-eye lens are used for the omnidirectional image acquisition. Further the following sensors are installed: ultrasound, infrared, a LIDAR, developed within a STWW project[2], a gyroscope, and an accelerometer. Two PCMCIA cards (DAQ-700 of National Instruments) are used to interface the sensors. Software is written in C++ using the ACE library[3]. A watchdog is implemented in hardware to detect when the software crashes. A wireless emergency button enables us to switch the power off at a distance, should this be required.

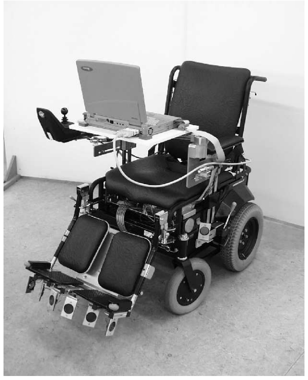

Figure 6: *The wheel chair Sharioto.*

5 Preliminary experimental results

In this section we will present an overview of the realized building blocks of the system so far. All computations were done on a 750 Mhz Pentium III machine operating under Linux.

The example pictures (color, resolution 640x480) are taken in a traffic-free shopping street in Leuven, Belgium. During a walk through this environment, we took 163 omnidirectional images with an interdistance of 2 meters approximately. Figure 9 shows the training tour on a map. Round red spots are image locations. Some image numbers are added to show the order. The images with numbers 20, 31, 88 and 154 are shown in figure 7.

5.1 Invariant region extraction

For each image in the set, invariant regions and their descriptors were computed. On the used system, this process took an average time per image of 9.42

[2] Sensor-assisted wheel chair control featuring shared autonomy, STWW project sponsored by the Flemish government and coordinated by IWT, http://www.mech.kuleuven.ac.be/pma/project/stww/

[3] Umar Syyid, The Adaptive Communication Environment ACE, A tutorial, Hughes Network Systems 1998, http://www.cs.wustl.edu/∼schmidt/

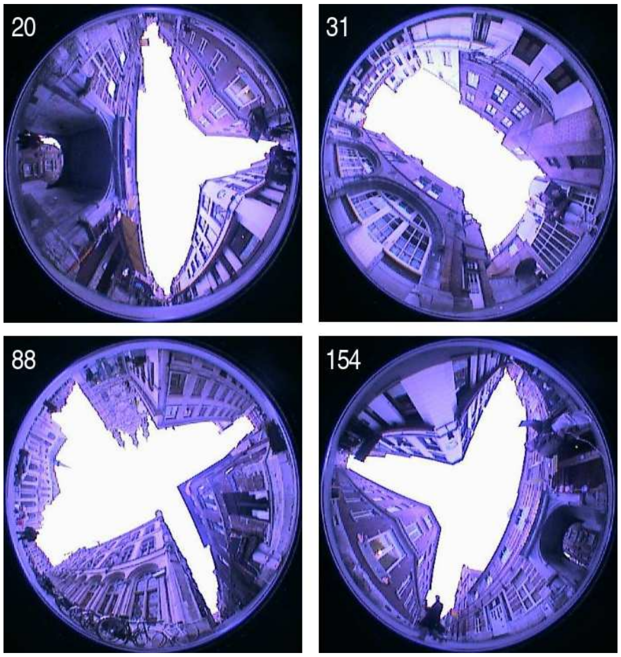

Figure 7: *Images 20, 31, 88 and 154 of the test set.*

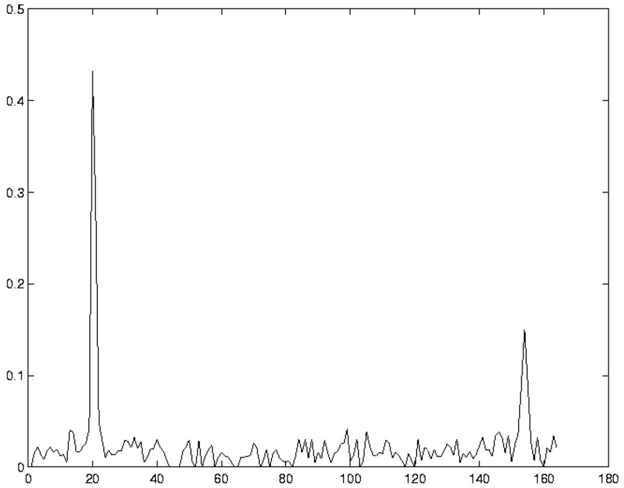

Figure 8: *Similarity measure between image 20 and the remaining images in the test set.*

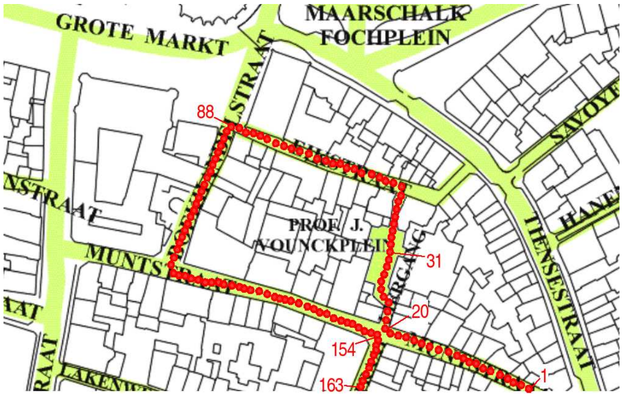

Figure 9: *A map of the training tour through a shopping street. Round red spots are image locations.*

seconds. The average number of regions detected in one image was 167. In figure 1 the parallelograms delineate the found affine regions.

5.2 Regions matching with database indexing

Out of these regions a database was built. By adding an index to this database, the search time is very low. For details of this algorithm, see [15]. The computation of the matches for one query image with all the remaining 162 images in the database took about 0.8 seconds.

5.3 Topological map building

For each pair of images, the similarity measure was computed. This resulted in a graph representation, which is too complex to show here. As an example, we show the similarity measure between image 20 and the remaining database images in figure 8. Notice the peak near image 154 which indicates the resemblance of these two images. In figure 9 it is clear that these images are taken at approximately the same place.

6 Summary and Conclusions

This paper presented results of an ongoing research project on the fusion of visual and inertial sensors for intelligent wheel chair control. The aim is building a world map that reflects the topological structure of the environment. First results on solely vision based tests show the power of this approach. A future combination with inertial sensors seems promising.

Acknowledgments

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). The scientific responsibility is assumed by the authors.

References

[1] M. Nuttin, E. Demeester, D. Vanhooydonck, H. Van Brussel, *Obstacle Avoidance and Shared Autonomy for Electric Wheel Chairs: An Evaluation of Preliminary Experiments*, Robotik, Germany, June 2002.

[2] M. Nuttin, D. Vanhooydonck, E. Demeester, H. Van Brussel, *Selection of suitable human-robot interaction techniques for intelligent wheelchairs*, Proc. of 11th IEEE International Workshop on Robot and Human Interactive Communication ROMAN, pp. 146-151, 2002.

[3] K. Dorfmüller, *Robust tracking for augmented reality using retroreflective markers*, Computers & Graphics 23, pp.795-800, 1999

[4] S. Thompson, T. Matsui, and A. Zelinsky, *Localization using Automatically Selected Landmarks from Panoramic Images*, Proceedings of Australian Conference on Robotics and Automation (ACRA2000), 2000.

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

[6] Z. Zhang, *Estimating motion and structure from correspondences of line segments between two perspective images*, IEEE Transaction on Pattern Analysis and Machine Intelligence, pp. 1129-1139, 1995.

[7] C. Schmidt and R. Mohr, *Local greyvalue invariants for image retrieval*, IEEE Transaction on Pattern Analysis and Machine Intelligence, pp. 872-877, 1997.

[8] A. Baumberg, *Reliable feature matching across widely separated views*, ICRA, pp. 774-781, 2000.

[9] F. Schaffalitzky and A. Zisserman, *Viewpoint invariant texture matching and wide baseline stereo*, ICCV, pp. 636-643, 2001.

[10] J. Shi and C. Tomasi, *Good features to track*, IEEE Conf. on Computer Vision and Pattern Recognition, pp.593-600, Seattle, June 1994

[11] D. Tell, *Wide Baseline Matching with applications to Visual Servoing,* doctoral dissertation, University Stockholm, 2002.

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

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

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

[15] H. Shao, T. Svoboda, T. Tuytelaars, and L. Van Gool, *HPAT indexing for fast object/scene recognition based on local appearance*, In proc. of the International Conference on Image and Video Retrieval (ICIVR), 2003.

[16] S. Se, D. Lowe and J.J. Little, *Vision-based mobile robot localization and mapping using scale-invariant features*, ICRA, pp. 2051-2058, 2001.

[17] S. Livatino and C.B. Madsen, *Autonomous Robot Navigation with Automatic Learning of Visual Landmarks*, Symposium on Intelligent Robotic Systems, 1999.

[18] V. Ferrari, T. Tuytelaars and L. Van Gool, *Markerless Augmented Reality with a Real-time Affine Region Tracker*, Proc. IEEE and ACM Intl. Symposium on Augmented Reality, volume I, pages 87-96, 2001.

[19] A. Davison and D. Murray, *Mobile Robot Navigation Using Active Vision*, ECCV, Vol. II, pp. 809-825, 1998.

[20] I. Ulrich and Illah Nourbakhsh, *Appearance-Based Place Recognition for Topological Localization*, IEEE International Conference on Robotics and Automation, pp. 1023-1029, 2000.

[21] M.O. Franz, B. Schölkopf, Ph. Georg, H.A. Mallot, and H.H. Bülthoff, *Learning View Graphs for Robot Navigation*, In Proc. 1. Intl. Conf. on Autonomous Agents, 1997.

[22] H.-M. Gross, A. König, H.-J. Böhme, and Ch. Schröter, *Vision-based Monte Carlo Self-localization for a Mobile Service Robot Acting as Shopping Assistant in a Home Store*, Proc. Int. Conference on Intelligent Robots and Systems, 2002.

[23] J. Borenstein, H.E. Everett, L. Feng, and D. Wehe, *Mobile Robot Positioning — Sensors and Techniques*, Journal of Robotic Systems, Special Issue on Mobile Robots, Vol. 14 No. 4, pp. 231-249.

[24] S. You, U. Neumann, R. Azuma, *Hybrid Inertial and Vision Tracking for Augmented Reality Registration*, Proceedings of IEEE VR '99 (Houston, TX, 13-17 March 1999).