53 Vision-Based Topological Navigation: An Implicit Solution to Loop Closure

Youcef Mezouar[1,3] · *Jonathan Courbon*[1,3] · *Philippe Martinet*[2,3]
[1]Clermont Université, Université Blaise Pascal, LASMEA
[2]Clermont Université, IFMA, LASMEA
[3]CNRS, LASMEA

A. Eskandarian (ed.), *Handbook of Intelligent Vehicles*, DOI 10.1007/978-0-85729-085-4_53,
© Springer-Verlag London Ltd. 2012

Abstract: Autonomous navigation using a single camera is a challenging and active field of research. Among the different approaches, visual memory-based navigation strategies have gained increasing interests in the last few years. They consist of representing the mobile robot environment with visual features topologically organized gathered in a database (visual memory). Basically, the navigation process from a visual memory can be split in three stages: (1) visual memory acquisition, (2) initial localization, and (3) path planning and following (refer to ➋ *Fig. 53.1*). Importantly, this frame work allows accurate autonomous navigation without using explicitly a loop closure strategy. The goal of this chapter is to provide to the reader an illustrative example of such a strategy.

1 Overview

Visual memory-based topological navigation refers to the use of prerecorded and topologically organized 2D image data to drive a robot along a learned trajectory. It relies on techniques inspired from visual servo controls. A major advantage of visual servo control is that absolute geometrical localization of the robot is not required to achieve positioning tasks and thus that drift errors are not propagated along the robot trajectory. However, the use of visual servo control in the field of autonomous navigation faces two major problems: (1) the robot is prone to large displacements which implies that current visual data cannot necessarily be matched with the reference data; (2) conventional visual servo controls make the assumption that a diffeomorphism between the image space and the robot's configuration space exists. Due to the nonholomic constraints of most of wheeled mobile robots, under the condition of rolling without slipping, such a diffeomorphism does not exist if the camera is rigidly fixed to the robot. A potential solution to the first of these two problems is to exploit a suitable environment representation (called visual memory in the sequel) allowing a description of the navigation task as a set of subgoals specified in the observation space. The second problem is often circumvented by providing extra degrees of freedom to the visual sensor. The goal of this chapter is to provide a complete and illustrative framework allowing visual memory-based navigation of non-holonomic wheeled mobile robots without adding extra DoFs to the camera.

The authors of (DeSouza and Kak 2002) account for 20 years of work at the intersection between the robotics and computer vision communities. In many works, as in (Hayet et al. 2002), computer vision techniques are used in a landmark-based framework. Identifying extracted landmarks with known reference points allows to update the results of the localization algorithm. These methods are based on some knowledge about the environment, such as a given 3D model or a map built online. They generally rely on a complete or partial 3D reconstruction of the observed environment through the analysis of data collected from disparate sensors. The vehicle can thus be localized in an absolute reference frame. Both motion planning and vehicle control can then be designed in this space. The results obtained by the authors of (Royer et al. 2007) leave to be forecasted that

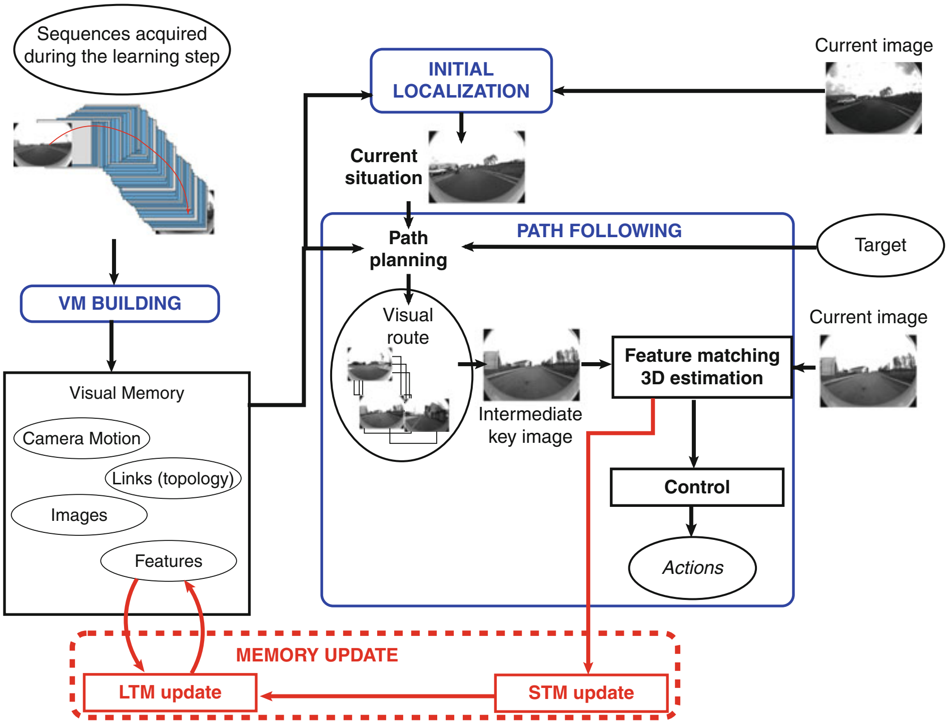


Fig. 53.1
Navigation process from a visual memory

such a framework will be reachable using a single camera. However, although an accurate global localization is unquestionably useful, the aim of this chapter is to present an alternative to build a complete vision-based framework without recovering the position of the vehicle with respect to a reference frame.

Visual memory-based navigation approaches have gained increasing interest in the last few years. They consist of representing the mobile robot environment with visual features gathered in a database (visual memory). Basically, the navigation process from a visual memory can be split in three stages: (1) visual memory acquisition, (2) initial localization, and (3) path following (refer to ➋ *Fig. 53.1*). In the first stage, a sequence of images is acquired, generally during a supervised step, and the robot's internal representation of the environment is built. Basically, three classes of internal representation can be distinguished (DeSouza and Kak 2002): map-less representation, topological and metrical maps. In (Matsumoto et al. 1996), a sequence of images, called *view-sequenced route reference*, is stored in the robot's *brain* for future navigation tasks. Such an approach is ranked among *map-less* as any notion of map or topology of the environment appears, neither to build the reference set of images, nor for the automatic guidance of the mobile robot. More classically, the visual memory is represented by a topological or a metrical map. In the first case, the nodes of the topological graph represent generally distinctive places while the edges denote connectivity between the places. In metrical maps, the visual memory consists more often of an accurate and consistent 3D representation of the environment. Structure-from-Motion (SfM; Nistér 2004; Royer et al. 2007) and Visual Simultaneous Localization and Mapping (V-SLAM; Lemaire et al. 2007) techniques can be used to build this representation. The SfM problem consists of retrieving the structure of the scene and the motion of the camera using the relation between the views and the correspondences between the features. The number of images of the video sequence initially acquired may be very large and the camera displacement between two views (*baseline*) is however often limited which makes the computation of matching tensors (such as the fundamental matrix) ill conditioned. A solution to decrease this problem is to select a subset of images (*key frames*). Many ways to choose those key images have been proposed (Torr 2002; Pollefeys et al. 2004; Thormählen et al. 2004), balancing the baseline and the number of matched points. Once the key images are chosen, these views, image points, and matched keypoints between successive images can be added to the visual memory. The whole structure of the environment may be built afterward using sequential SfM. Two or three views are usually used to retrieve a first seed 3D structure (Pollefeys et al. 2004; Nistér 2004). Key frames are then sequentially added, computing the pose of each new camera using the previously estimated 3D points (*resection* step). Subsequently, the 3D structure is updated by triangulating the 3D points conveyed by the new view. Both structure and motion are optimized using global (as in Triggs et al. 2000) or local (as in Mouragnon et al. 2009) bundle adjustment. The output of this learning process is a 3D reconstruction of the scene which contains the pose of the camera for each key image and a set of 3D points associated with interest points. The SLAM problem consists of the estimation of the observed environment feature location (*mapping*) and of the robot's pose (*localization*), two problems intimately tied together. Stochastic approaches have proved to solve the SLAM problem in a consistent way because they explicitly deal with sensor noise. A feature-based SLAM approach generally encompasses four basic functionalities: feature selection, relative measures estimation, data association, and estimation. In V-SLAM, the observed features can be for instance interest points detected in the images and data association performed by a feature matching process. Filters like the Extended Kalman Filter are then used to estimate both the localization of the robot and the 3D position of features on the environment. The second stage of the navigation process (initial localization) consists of finding the position of the robot in its internal representation of the environment using the current image acquired by the embedded camera. It can rely on image matching and/or on the matching of features extracted from the current image and images stored in the visual memory. Once the robot is localized and a target is specified in its internal representation of the environment, the next stage (navigation) consists first in planning the robot's mission and second to perform it autonomously. In the sequel, this chapter will focus on navigation strategies where key images are stored in the visual memory and are used as references during the online steps.

2 Environment Representation

In (DeSouza and Kak 2002), approaches using a "memorization" of images of the environment acquired with an embedded camera are ranked among map-less navigation systems. As proposed in (Matsumoto et al. 1996) or (Jones et al. 1997), neither notion of mapping nor topology of the environment appears, in building the reference set of images, nor for the automatic guidance of the vehicle. The first step in vision-based topological navigation strategies consists of a learning stage to build the visual memory.

2.1 Visual Memory Structure

The environment is supposed to contain a set of 3D features $\{Q_l \mid l = 1, 2, \ldots n\}$. The observation (or projection) of a 3D feature Q_l in an image $\mathcal{I}^{i_a}$ is a visual feature noted $\mathcal{P}_l^*$ (refer to ❯ *Fig. 53.2*). It is assumed that visual features can be located/detected from images and that they are described by feature vectors. Two features $\mathcal{P}_{l_1}^{i1}$ and $\mathcal{P}_{l_2}^{i2}$ from two images I^{i1} and I^{i2} are said to be *matched* or *in correspondence* if they are supposed to be the projections of a same 3D feature (i.e., $l_1 = l_2$).

2.1.1 Visual Memory

The visual memory of the robot can store different features. In this chapter, the concept of visual memory is illustrated assuming that the following 2D features are stored:

(a) n_{VM} key images $\{I^i \mid i = \{1, 2, \ldots, n_{VM}\}\}$ extracted from a video sequence
(b) For each key image I^i, a set P^i of n^i descriptive image features $P^i = \left\{\mathcal{P}_{l_j}^i | j = \{1, 2, \ldots, n^i\},\ l_j \in \{1, 2, \ldots n\}\right\}$
(c) A set of links between adjacent places $\left\{\left(\mathcal{I}^{i_a}, \mathcal{I}^{i_b}\right), (i_a, i_b) \in \{1, 2, \ldots, n_{VM}\}^2, i_a \neq i_b\right\}$

2.1.2 Visual Paths

A visual path Ψ^p is a weighted directed graph composed of n successive key images (*vertices*):

$$\Psi^p = \left\{\mathcal{I}_i^p | i \in \{1, 2, \ldots, n\}\right\}$$

For control purpose (refer to ❯ Sect. 4), the authorized motions during the learning stage are assumed to be limited to those of a car-like vehicle, which only goes forward. The following Hypothesis 1 formalizes these constraints.

Hypothesis 1: Given two frames ${}^R\mathcal{F}_i$ and ${}^R\mathcal{F}_{i+1}$, respectively associated to the vehicle when two successive key images I_i and I_{i+1} of a visual path Ψ were acquired, there exists an

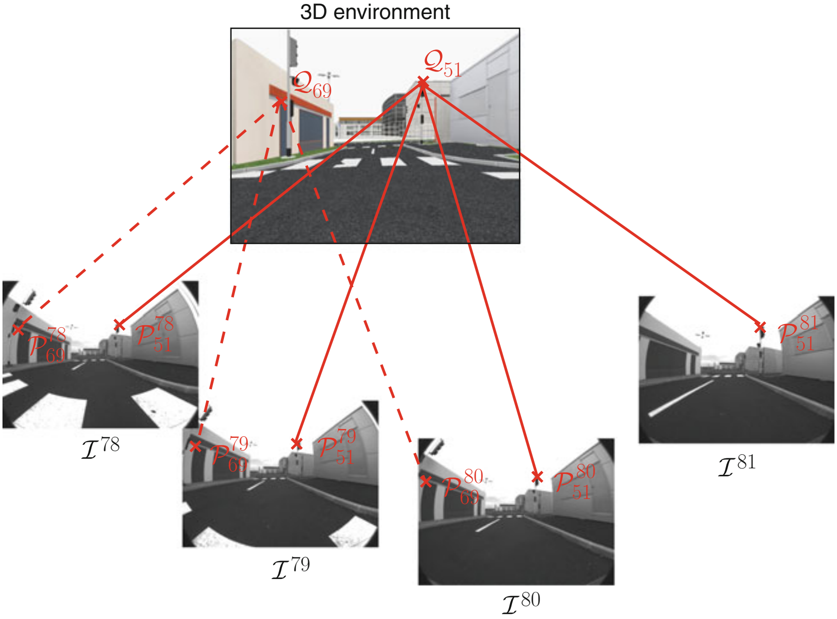


Fig. 53.2
Images, 3D features, and visual features

admissible path ψ from ${}^{R}\mathcal{F}_i$ to ${}^{R}\mathcal{F}_{i+1}$ for a car-like vehicle whose turn radius is bounded, and which only moves forward.

Moreover, because the controller is assumed vision based, the vehicle is controllable from I_i to I_{i+1} only if the hereunder Hypothesis 2 is respected.

Hypothesis 2: Two successive key images I_i and I_{i+1} contain a set P_i of matched visual features, which can be observed along a path performed between ${}^{R}\mathcal{F}_i$ and ${}^{R}\mathcal{F}_{i+1}$ and which allows the computation of the control law.

In the sequel, this chapter is illustrated using interest points as visual features. During the acquisition of a visual path, the Hypothesis 2 constrains the choice of the key images. As a consequence of Hypothesis 1 and 2, each visual path Ψ^p corresponds to an oriented edge which connects two configurations of the vehicle's workspace. The *weight of a visual path* can be defined for instance as its cardinal.

2.1.3 Visual Memory Vertices

In order to connect two visual paths, the terminal extremity of one of them and the initial extremity of the other one must be constrained as two consecutive key images of a visual path. The paths are then connected by a vertex, and two adjacent vertices of the visual memory are connected by a visual path.

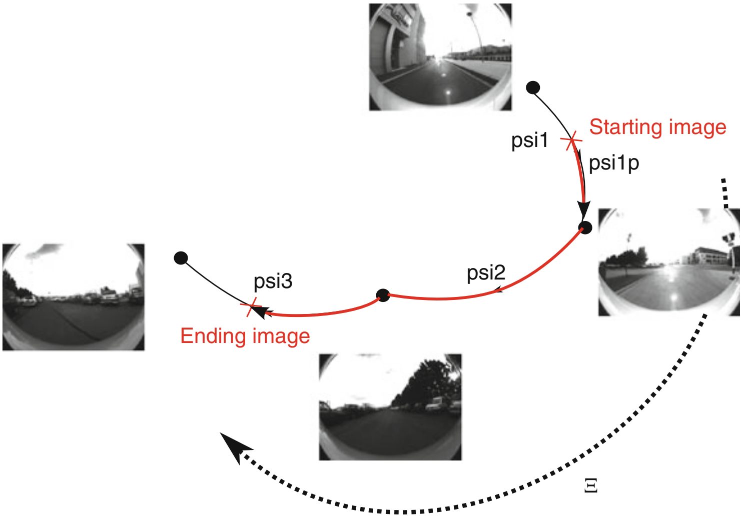
Proposition 1: Given two visual paths $\Psi^{p_1} = \left\{\mathcal{I}_i^{p1} | i \in \{1, 2, \ldots, n_1\}\right\}$ and $\Psi^{p_2} = \left\{\mathcal{I}_i^{p2} | i \in \{1, 2, \ldots, n_2\}\right\}$, if the two key images $\mathcal{I}_{n_1}^{p_1}$ and $\mathcal{I}_1^{p_2}$ abide by both Hypothesis 1 and 2, then a vertex connects Ψ^{p_1} to Ψ^{p_2}.

2.1.4 A Connected Multigraph of Weighted Directed Graphs

According to ➋ Sects. 2.1.2 and ➋ 2.1.3, the visual memory structure is a multigraph in which vertices are key images linked by edges which are the visual paths (*directed graphs*). Note that more than one visual path may be incident to a node. It is yet necessary that this multigraph is strongly connected. This condition guarantees that any vertex of the visual memory is attainable from every other, through a set of visual path.

2.2 Visual Route

A visual route describes the vehicle's mission in the sensor space. Given two key images of the visual memory $\mathcal{I}_s^*$ and I_g, corresponding respectively to the starting and goal locations of the vehicle in the memory, a visual route is a set of key images which describes a path from $\mathcal{I}_s^*$ to I_g, as presented in ➋ *Fig. 53.3*. $\mathcal{I}_s^*$ is the closest key image to the current image I_s. The image $\mathcal{I}_s^*$ is extracted from the visual memory during a localization step. The visual route can be chosen for instance as the minimum length path of the visual memory connecting two vertices associated to $\mathcal{I}_s^*$ and I_g. According to the definition of the



■ **Fig. 53.3**
The tasks consists of navigating from the starting to the ending images. With this aim, a visual route $\Xi = \Psi^{1'} \oplus \Psi^2 \oplus \Psi^{3'}$ connecting these two images is defined

value of a visual path, the length of a path is the sum of the values of its arcs. Consequently, the visual route results from the concatenation of indexed visual paths. Given two visual paths Ψ^{p_1} and Ψ^{p_2}, respectively containing n_1 and n_2 indexed key images, the concatenation operation of Ψ^{p_1} and Ψ^{p_2} is defined as follows:

$$\Psi^{p_1} \oplus \Psi^{p_2} = \left\{\mathcal{I}_j^{p_{1,2}} | j = \{1, \ldots, n_1, n_1 + 1, \ldots, n_1 + n_2\}\right\}$$

$$\mathcal{I}_j^{p_{1,2}} = \begin{cases} \mathcal{I}_j^{p_1} & \text{if } j \leq n_1 \\ \mathcal{I}_{j-n_1}^{p_2} & \text{if } n_1 < j \leq n_1 + n_2 \end{cases}$$

2.3 Key Images Selection

A central clue for implementation of this framework relies on efficient point matching. It allows key image selection during the learning stage, of course it is also useful during autonomous navigation in order to provide the necessary input for state estimation. A simple but efficient solution to this issue is given in (Royer et al. 2007) and was successfully applied for the metric localization of autonomous vehicles in outdoor environment. Interest points are detected in each image with Harris corner detector (Harris and Stephens 1988). For an interest point $\mathcal{P}_1$ at coordinates $(x\ y)$ in image I_i, a search region in image I_{i+1} is defined. For each interest point $\mathcal{P}_2$ inside the search region in image I_{i+1}, a similarity score is computed between the neighborhoods of $\mathcal{P}_1$ and $\mathcal{P}_2$ using a zero-normalized cross correlation. The point with the best score is kept as a good match and the unicity constraint is used to reject matches which have become impossible. This method is illumination invariant and its computational cost is small. The first image of the video sequence is selected as the first key frame I_1. A key frame I_{i+1} is then chosen so that there are as many video frames as possible between I_i and I_{i+1} while there are at least M common interest points tracked between I_i and I_{i+1}.

2.4 Visual Memory Update

The internal representation of the environment is generally built once and never changed. Most navigation strategies proposed in the literature assume that the environment where the robot works is static. However, this assumption does not hold for many real environments. Following the taxonomy proposed in (Yamauchi and Langley 1997), changes in the environment may be *transient* or *lasting*. Transient changes are brief enough and can be handled reactively. In general, it does not require any long-standing modification of the robot's internal memory (for instance, moving objects or walking pedestrians). Lasting changes persist over longer periods of time and have to be memorized by the robot. They may be *topological* (changes in the topology) and/or *perceptual* (changes in the appearance of the environment).

As noted previously, perceptual lasting changes will deteriorate the feature-matching process and then the performance of vision-based navigation strategies. To improve the navigation performances, new lasting features have to be incorporated in the map of the environment. Further, obsolete elements have to be eliminated to limit the required resources in terms of memory and processing power over time.

As mentioned previously, a large part of the literature deals with transient changes. The robot's environment is generally decomposed into a static part and a dynamic part encapsulating ephemeral (potentially moving) objects. Two solutions can be used to deal with this situation. The first solution consists of identifying the parts of the environment which are not consistent with a predefined static model. This is usually bypassed with geometric consistency of view matching. The second solution consists of tracking moving objects as proposed in the context of V-SLAM in (Bibby and Reid 2007; Wangsiripitak and Murray 2009). These objects can then be integrated to the map building process as in (Bibby and Reid 2007) or rejected as in (Wangsiripitak and Murray 2009). However, these solutions may improve the current localization but cannot handle long-term changes on the structure of the environment.

Only few works have been devoted to lasting changes. In feature-based visual SLAM approaches, features accumulate over time (which can be seen as a map update) but obsolete features are not discarded. It results a growing of the required memory and processing power over time and an efficiency loss. In (Hochdorfer and Schlegel 2009), the evaluation of the quality of the localization allows to rank landmarks and to eliminate less useful ones. In (Andreasson et al. 2007), the initial map is supposed to be partially correct and a robust method for global place recognition in scenes subject to changes over long periods of time is proposed. As the reference view is never modified, this approach may be inefficient after some times. It seems more promising to modify the reference views as proposed in (Dayoub and Duckett 2008; Dayoub et al. G 2010; Bacca et al. 2010) for localization. The information model used in those works is based on the human memory model proposed in (Atkinson and Shiffrin 1968) and the concepts of short-term and long-term memories. Basically, reference views are stored in a long-term memory (LTM). When features have been seen in many views during the localization step, they are transferred from the short-term memory (STM) to the long-term memory (if they do not belong yet to it) and missing features are forgotten (and are deleted after sometime). The updates of the memories are based on a finite state machine in (Dayoub and Duckett 2008; Dayoub et al. 2010) and on feature stability histograms built using a voting scheme in (Bacca et al. 2010). Those approaches are tested with images acquired by omnidirectional cameras in indoor environments. It is reported that localization performances are improved with respect to a static map.

3 Localization in a Visual Memory

The output of the learning process is a data set of images (*visual memory*). The first step in the autonomous navigation process is the self-localization of the vehicle in the visual memory. In a visual memory, the localization consists of finding the image of the memory which best fits the current image by comparing preprocessed and online acquired images. Two main strategies exist to match images: The image can be represented by a single descriptor (global approaches) (Matsumoto et al. 1999; Linåker Fand and Ishikawa 2004) or alternatively by a set of descriptors defined around visual features (landmark-based or local approaches) (Goedemé et al. 2005; Tamimi et al. 2005; Murillo et al. 2007). Some hybrid approaches based on a global description of a subset of the image have also been proposed to increase the robustness of global methods (Gonzalez-Barbosa and Lacroix 2002). On the one hand, local approaches are generally more accurate but have a high computational cost (Murillo et al. 2007). On the other hand, global descriptors speed up the matching process at the price of affecting the robustness to occlusions. One solution consists in using a hierarchical approach which combines the advantages of both methods (Menegatti et al. 2003). In a first step, global descriptors allow to select only some possible images and then, if necessary, local descriptors are used to keep the best image. This section briefly reviews global and local descriptors for localization in a visual memory with a particular focus on wide field-of-view images since they are of particular interests in the context of autonomous navigation.

3.1 Global Descriptors

A first solution is to globally describe the image. In that aim, images are mapped onto cylindrical images of size 128 × 32 in (Matsumoto et al. 1999). The image is directly described by the gray-level values. In (Pajdla Tand and Hlaváč 1999), a shift invariant representation is computed by rotating the cylindrical image in a reference direction. Unfortunately, this direction is not absolute as soon as occlusions appear. In order to decrease the size of the memorized data, images can be represented by their eigenvectors using principal component analysis as proposed in (Gaspar et al. 2000). Unfortunately, when a new image is integrated in the memory, all eigenvectors have to be recomputed. This process is very complex and it has a very high computational cost. Moreover, those methods are not robust to changes of the environment. The histogram of the gray-level values is largely employed as global signature. Its computation is efficient and it is rotation-invariant. However, histogram methods are sensitive to change of light conditions. Blaer and Allen (2002) propose color histograms for outdoor scene localization. A normalization process is applied before computing the histograms in order to reduce the illumination variations. In (Linåker Fand and Ishikawa 2004), a global descriptor based on a polar version of high order local autocorrelation functions (PHLAC) is proposed. It is based on a set of 35 local masks applied to the image by convolution. Similar to histogram, this descriptor is rotation-invariant.

3.2 Local Descriptors

Global descriptor-based methods are generally less robust to occlusion compared to landmark-based methods. In those last methods, some relevant visual features are extracted from the images. A descriptor is then associated to each feature neighborhood. The robustness of the extraction and the invariance of the descriptor are one main issue to improve the matching process. Two main approaches can be distinguished. In the first category, the feature detection and description designed for images acquired by perspective cameras are directly employed with omnidirectional images. The second category takes the geometry of the sensor into account and thus uses operators designed for omnidirectional images. The most popular visual features used in the context of localization in an image database are projected points. However, projected lines can also be exploited as proposed in (Murillo et al. 2007).

1. *Perspective-based local descriptor*: The Scale Invariant Feature Transform (SIFT, (Lowe 2004)) has been shown to give the best results in the case of images acquired with perspective cameras. The SIFT descriptor is a set of histograms of gradient orientations of the normalized (with respect to orientation and scale) difference of Gaussian images. In view of the effectiveness of this descriptor, several extensions have been proposed. It has been used with omnidirectional images in (Goedemé et al. 2005). Given that many points are detected in an omnidirectional image, Tamimi et al. (2005) proposed an iterative SIFT with a lower computational cost. In (Andreasson et al. 2005), points are detected with a Sobel filter and described by a Modified Scale Invariant Feature Transform (M-SIFT) signature. This signature slightly takes into account the sensor geometry by rotating the patch around an interest point. In (Murillo et al. 2007), the Speeded-Up Robust Features (SURF) are employed as descriptors. SURF points are detected using the Hessian matrix of the image convolved with box filters and the descriptor is computed thanks to Haar wavelet extraction. The computational cost of this descriptor is much lower than the one obtained for SIFT. Unfortunately, those signatures describe a local neighborhood around interest points and do not take into account the high distortions caused by the sensor geometry.
2. *Descriptors adapted to wide-angle images*: In the second category, detection and description processes are specially designed to take into account high distortions. In (Svoboda and Pajdla 2001; Ieng et al. 2003), a classical Harris corner detector is proposed but the shape and the size of a patch around a feature is modified according to the position of the point and to the geometry of the catadioptric sensor. Finally, a standard 2D correlation (respectively a centered and normalized cross correlation) is applied to the patches in (Svoboda and Pajdla 2001) (respectively in Ieng et al. 2003). After computing the descriptors of the current and memorized images, those descriptors have to be matched. For local approaches, this step is generally based on pyramidal matching as in (Murillo et al. 2007) or on nearest neighbor matching as in (Lowe 2004). This last algorithm considers that a matching is correct if the ratio between the distances of the first and second nearest neighbors is below a threshold. It is possible to eliminate wrong matching through the recovery of the epipolar geometry between two views (Zhang et al. 1995) at the price of higher computational cost. A full reconstruction can also be obtained with three views and the 1D trifocal tensor as proposed in (Murillo et al. 2007).

3.3 Hybrid Descriptors

Some hybrid descriptors have been designed to combine the advantages of the two previously cited categories (local and global approaches) by globally describing subsets of the image. In (Gonzalez-Barbosa and Lacroix 2002), five histograms of the first- and second-order derivatives of the gray-level image are considered. Instead of the whole image, the image is decomposed into rings. On the one hand, a decomposition into few rings decreases the accuracy. On the other hand, increasing the number of rings increases the computational cost and decreases the robustness to occlusions. In (Gaspar et al. 2000), the image is first projected onto an englobing cylinder and a grid decomposition is then proposed. This projection step is time consuming and it implies the modification of the quality of the image which can lead to less accurate localization results. In (Courbon et al. 2008), a hierarchical process combining global descriptors computed onto cubic interpolation of triangular mesh and patches correlation around Harris corners has been proposed. In the context of visual memory-based navigation, this method has shown the best compromise in terms of accuracy, amount of memorized data required per image, and computational cost (refer to (Courbon et al. 2008) for detailed results).

4 Route Following

Given an image of one of the visual paths as a target, the navigation task in a visual memory-based framework can formally be defined as the regulation of successive error functions allowing the guidance of the robot along the reference visual route. The visual route describes then a set of consecutive states that the image has to reach in order that the robot joins the goal configuration from the initial one. Control schemes suitable in this context can be designed by exploiting visual-servoing concepts. Visual servoing is often considered as a way to achieve positioning tasks. Classical methods, based on the task function formalism, make the assumption that a diffeomorphism between the sensor space and the robot's configuration space exists. Due to the nonholomic constraints of most of wheeled mobile robots, under the condition of rolling without slipping, such a diffeomorphism does not exist if the camera is rigidly fixed to the robot. In (Tsakiris et al. 1998), the authors add extra degrees of freedom to the camera. The camera pose can then be regulated in a closed loop. In the case of an embedded and fixed camera, the control of the camera is generally based on wheeled mobile robots control theory (Samson 1995). In (Ma et al. 1999), a car-like robot is controlled with respect to the projection of a ground curve in the image plane. The control law is formalized as a path-following problem. More recently, in (Fang et al. 2002) and (Chen et al. 2003), a partial estimation of the camera displacement between the current and desired views has been exploited to design vision-based control laws. The camera displacement is estimated by uncoupling translation and rotation components of an homography matrix. In (Fang et al. 2002), a time-varying control allows an asymptotical stabilization on a desired image. In (Chen et al. 2003), a trajectory-following task is achieved. The trajectory to follow is defined by a prerecorded video and the control law is proved stable using Lyapunov-based analysis. In (Goedemé et al. 2005), homing strategy is used to control a wheelchair from a memory of omnidirectional images. A memory of omnidirectional images is also used in (Gaspar et al. 2000) where localization and navigation are realized in the bird's-eye (orthographic) views obtained by radial distortion correction of the omnidirectional images. The control of the robot is formulated in the bird's-eye view of the ground plane which is similar to a navigation in a metric map. The view-sequenced route presented in (Matsumoto et al. 1996) has been applied to omnidirectional images in (Matsumoto et al. 1999). The control scheme exploits the inputs extracted from unwarped images. For completeness, the control strategy proposed in (Courbon et al. 2009) to follow a visual route with a non-holonomic vehicle is briefly presented more in details.

The localization step provides the closest image $\mathcal{I}_s^*$ to the current initial image I_c. A visual route Ψ connecting $\mathcal{I}_s^*$ to the goal image can then be extracted from the visual memory. The principle of the vision-based control scheme is presented in ➲ *Fig. 53.4*.

4.1 Model and Assumptions

1. *Control objective*: Let I_i and I_{i+1} be two consecutive key images of a given visual route to follow and I_c be the current image. $\mathcal{F}_i = (O_i, \mathbf{X_i}, \mathbf{Y_i}, \mathbf{Z_i})$ and $\mathcal{F}_{i+1} = (O_{i+1}, \mathbf{X_{i+1}}, \mathbf{Y_{i+1}}, \mathbf{Z_{i+1}})$ are the frames attached to the vehicle when I_i and I_{i+1} were stored and $\mathcal{F}_c = (O_c, \mathbf{X_c}, \mathbf{Y_c}, \mathbf{Z_c})$ is a frame attached to the vehicle in its current location. ➲ *Figure 53.5* illustrates this setup. The origin O_c of $\mathcal{F}_c$ is on the center rear axle of a car-like vehicle, which moves on a perfect ground plane. The hand–eye parameters (i.e., the rigid transformation between $\mathcal{F}_c$ and the frame attached to the camera) are supposed to be known. According to Hypothesis 2, the state of a set of visual features $\mathcal{P}_i$ is known in the images I_i and I_{i+1}. The state of $\mathcal{P}_i$ is also assumed available in I_c (i.e., $\mathcal{P}_i$ is in the camera field of view). The task to achieve is to drive the state of $\mathcal{P}_i$ from its current value to its value in I_{i+1}. In the following, Γ represents a path from $\mathcal{F}_i$ to $\mathcal{F}_{i+1}$. The control strategy consists in guiding I_c to I_{i+1} by regulating asymptotically the axle $\mathbf{Y}_c$ on Γ. The control objective is achieved if $\mathbf{Y}_c$ is regulated to Γ before the origin of $\mathcal{F}_c$ reaches the origin of $\mathcal{F}_{i+1}$.
2. *Vehicle modeling*: The vehicle is supposed to move on asphalt at rather slow speed. In this context, it appears quite natural to rely on a kinematic model, and to assume pure rolling and nonslipping at wheel–ground contact. In such cases, the vehicle modeling is commonly achieved for instance relying on the Ackermann's model, also named the bicycle model: the two front wheels located at the mid-distance between actual front wheels and actual rear wheels. In the sequel, the robot configuration is described with respect to the path Γ, rather than with respect to an absolute frame. As seen previously, the objective is that the vehicle follows a reference path Γ. To meet this objective, the following notations are introduced (see ➲ *Fig. 53.5*).

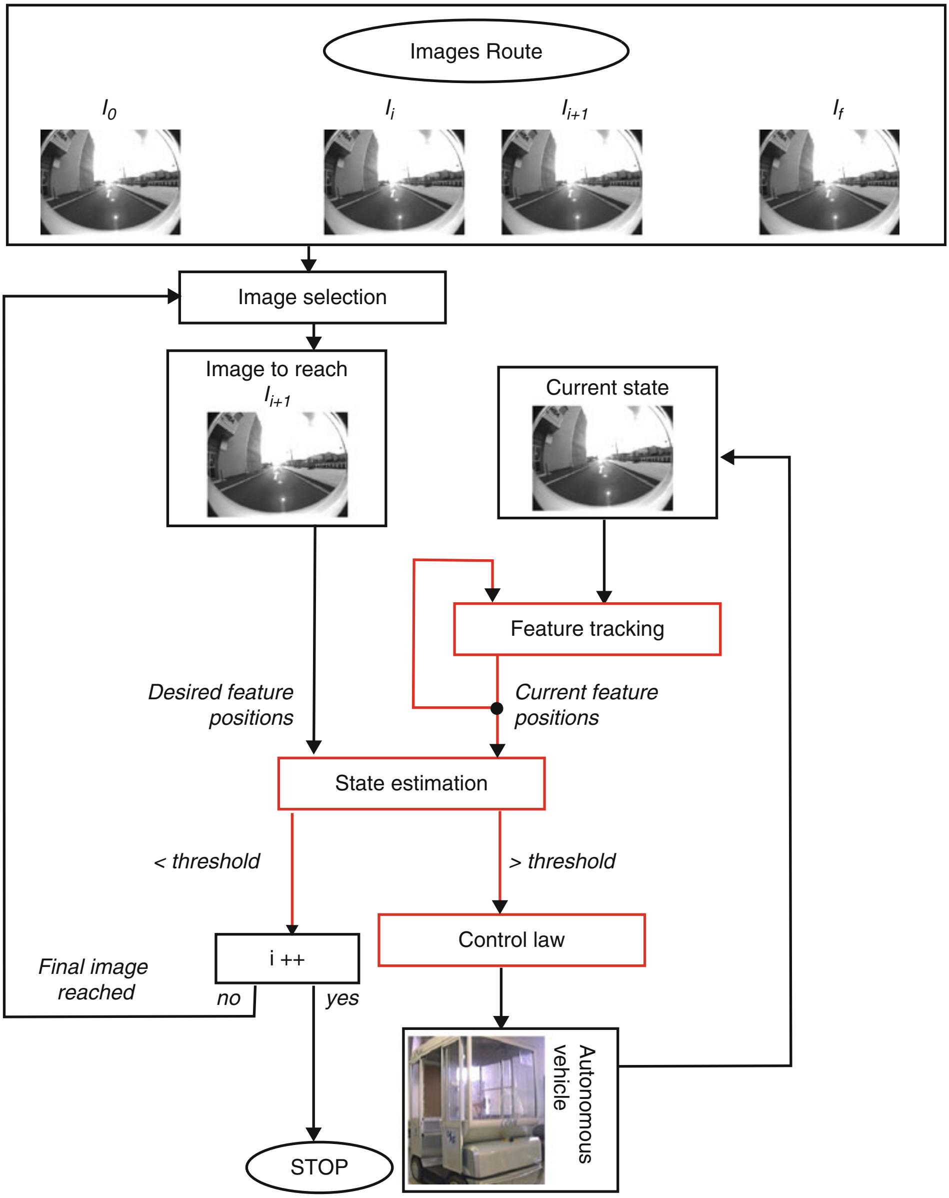


Fig. 53.4
Visual route following process

- O_C is the center of the vehicle rear axle.
- $\mathcal{M}$ is the point of Γ which is the closest to O_C. This point is assumed to be unique which is realistic when the vehicle remains close from Γ.
- s is the curvilinear coordinate of point M along Γ and $c(s)$ denotes the curvature of Γ at that point.

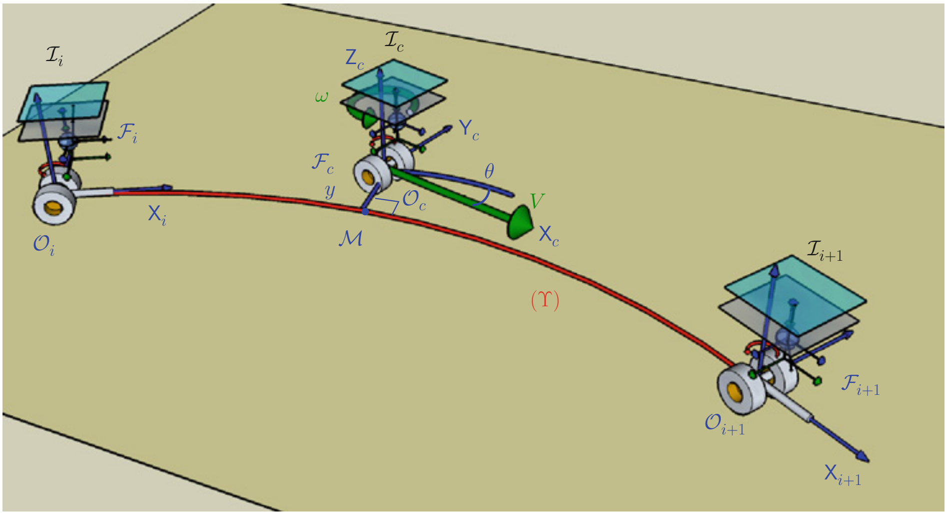


Fig. 53.5

Images $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$ are two consecutive key images of the visual route $\boldsymbol{\Psi}$. $\mathcal{I}_c$ is the current image. $\boldsymbol{\Gamma}$ is the path to follow

- y and θ are respectively the lateral and angular deviation of the vehicle with respect to reference path Γ.
- δ is the virtual front wheel steering angle.
- V is the linear velocity along the axle $\mathbf{Y_c}$ of $\mathcal{F}_c$.
- l is the vehicle wheelbase.

Vehicle configuration can be described without ambiguity by the state vector (s, y, θ): The two first variables provide point O_C location and the last one the vehicle heading. Since V is considered as a parameter, the only control variable available to achieve path following is δ. The vehicle kinematic model can then be derived by writing that velocity vectors at point O_C and at center of the front wheel are directed along wheel planes and that the vehicle motion is, at each instant, a rotation around an instantaneous rotation center. Such calculations lead to (refer to Zodiac 1995):

$$\begin{cases} \dot{s} = V \dfrac{\cos\theta}{1 - c(s)y} \\ \dot{y} = V \sin\theta \\ \dot{\theta} = V \left(\dfrac{\tan\delta}{l} - \dfrac{c(s)\cos\theta}{1 - c(s)y} \right) \end{cases} \tag{53.1}$$

Model (53.1) is clearly singular when $y = \frac{1}{c(s)}$, i.e., when point O_C is superposed with the path Γ curvature center at abscissa s. However, this configuration is never encountered in practical situations: On the one hand, the path curvature is small and on the other, the vehicle is expected to remain close to Γ.

4.2 Control Design

The control objective is to ensure the convergence of y and θ toward 0 before the origin of $\mathcal{F}_c$ reaches the origin of $\mathcal{F}_{i+1}$. The vehicle model (➤ 53.1) is clearly nonlinear. However, it has been established in (Samson 1995) that mobile robot models can generally be converted in an exact way into almost linear models, named chained forms. This property offers two very attractive features: On the one hand, path following control law can be designed and tuned according to Linear System Theory, while controlling nevertheless the actual nonlinear vehicle model. Control law convergence and performances are then guaranteed whatever the vehicle initial configuration is. On the other hand, chained form enables to specify, in a very natural way, control law in terms of distance covered by the vehicle, rather than in terms of time. Vehicle spacial trajectories can then easily be controlled, whatever the vehicle velocity is (Thuilot et al. 2004). Conversion of the vehicle model (➤ 53.1) into chained form can be achieved thanks to state and control transformations as detailed in (Thuilot et al. 2004) leading to the following expression of the control law:

$$\begin{aligned}\delta(y,\theta) = \arctan\Bigg(-l\Bigg[\frac{\cos^3\theta}{(1-c(s)y)^2}\Bigg(\frac{dc(s)}{ds}y\tan\theta - K_d(1-c(s)y)\tan\theta \\ - K_p y + c(s)(1-c(s)y)\tan^2\theta\Bigg) + \frac{c(s)\cos\theta}{1-c(s)y}\Bigg]\Bigg)\end{aligned} \tag{53.2}$$

The evolution of the error dynamics is driven by the distance covered by the vehicle along the reference path Γ). The gains (K_d, K_p) impose a settling distance instead of a settling time as it is usual. Consequently, for a given initial error, the vehicle trajectory will be identical, whatever the value of V is, and even if V is time varying ($V \neq 0$). Control law performances are therefore velocity independent. The gains (K_d, K_p) can be fixed for desired control performances with respect to a second-order differential equation. The path to follow can simply be defined as the straight line $\Gamma' = (O_{i+1}, \mathbf{Y}_{i+1})$ (refer to ➤ *Fig. 53.5*). In this case $c(s) = 0$ and the control law (➤ 53.2) can be simplified as follows:

$$\delta(y,\theta) = \arctan\big(-l\big[\cos^3\theta\big(-K_d\tan\theta - K_p y\big)\big]\big) \tag{53.3}$$

The implementation of control law (➤ 53.3) requires the online estimation of the lateral deviation y and the angular deviation θ of $\mathcal{F}_c$ with respect to Γ. In (Courbon et al. 2009) geometrical relationships between two views are exploited to enable a partial Euclidean reconstruction from which (y, θ) are derived.

5 Example of Results

5.1 Experimental Setup

The experimental vehicle is depicted in ➲ *Fig. 53.6*. It is an urban electric vehicle, named RobuCab, manufactured by the Robosoft Company. Currently, RobuCab serves as experimental testbed in several French laboratories. The 4 DC motors are powered by lead-acid batteries, providing 2 h autonomy. Vision and guidance algorithms are implemented in C^{++} language on a laptop using RTAI-Linux OS with a 2 GHz Centrino processor. The Fujinon fisheye lens, mounted onto a Marlin F131B camera, has a field of view of 185°. The image resolution in the experiments was 800 × 600 pixels. The camera, looking forward, is situated at approximately 80 cm from the ground. The parameters of the rigid transformation between the camera and the robot control frames are roughly estimated. Gray-level images are acquired at a rate of 15 fps. Two illustrative experiments are presented. The first one shows the loop closure performance while the second one shows that it is possible to achieve visual memory-based navigation in large environment.

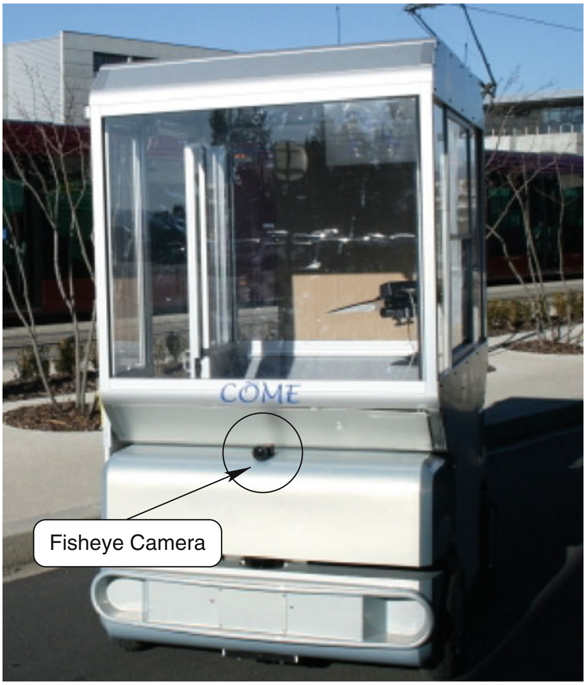


Fig. 53.6
RobuCab vehicle with the embedded camera

5.2 Loop Closure

Autonomous navigation along a loop is interesting because it is a good way to visualize the performances. Remarkably, the topological visual memory implicitly defines the loop closure. It is an advantage of this approach with respect to methods based on a metric representation of the environment which can be subject to significant drift if a loop closure process is not implicitly incorporated to the navigation strategy. The path is defined from the concatenation of the sequences Γ^1, Γ^2, and Γ^3. It is a 270 m loop (refer to ➋ *Fig. 53.7*). A total of 1,100 images were acquired and the resulting visual memory contains three sequences and 153 key frames. In this experiment, the navigation task consists in performing five consecutive loops. The results are given in ➋ *Fig. 53.8*. One can verify that the robot reaches the position corresponding to the first image of Γ^1 at the end of a "loop."

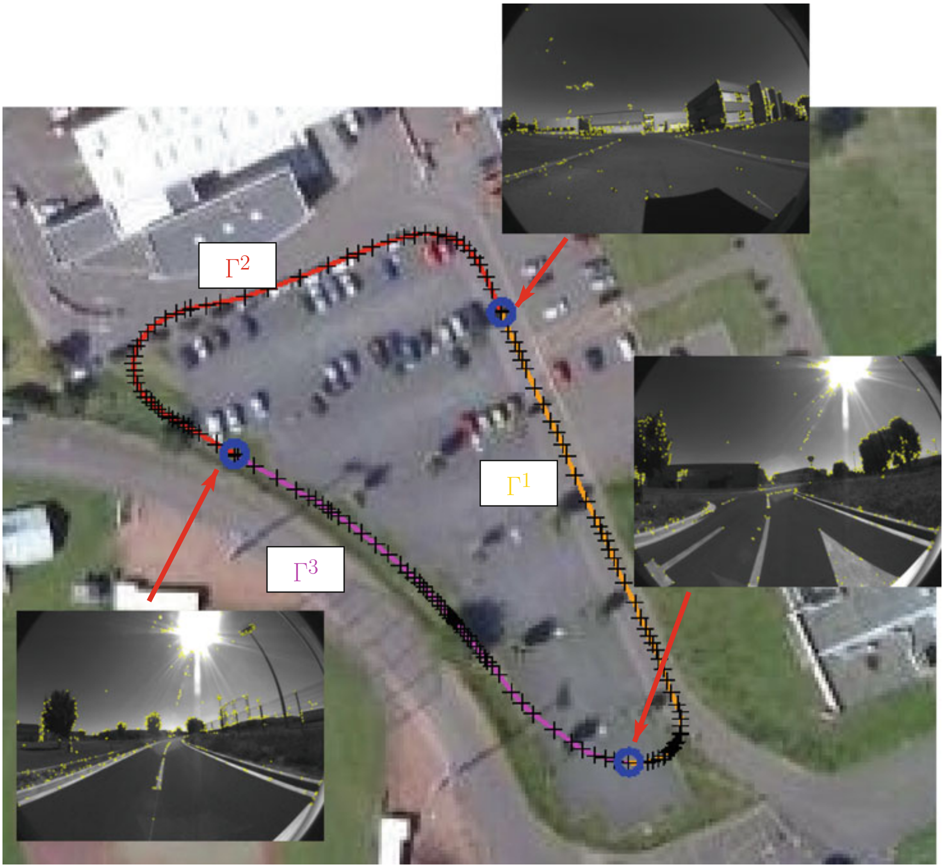


Fig. 53.7
The test Loop

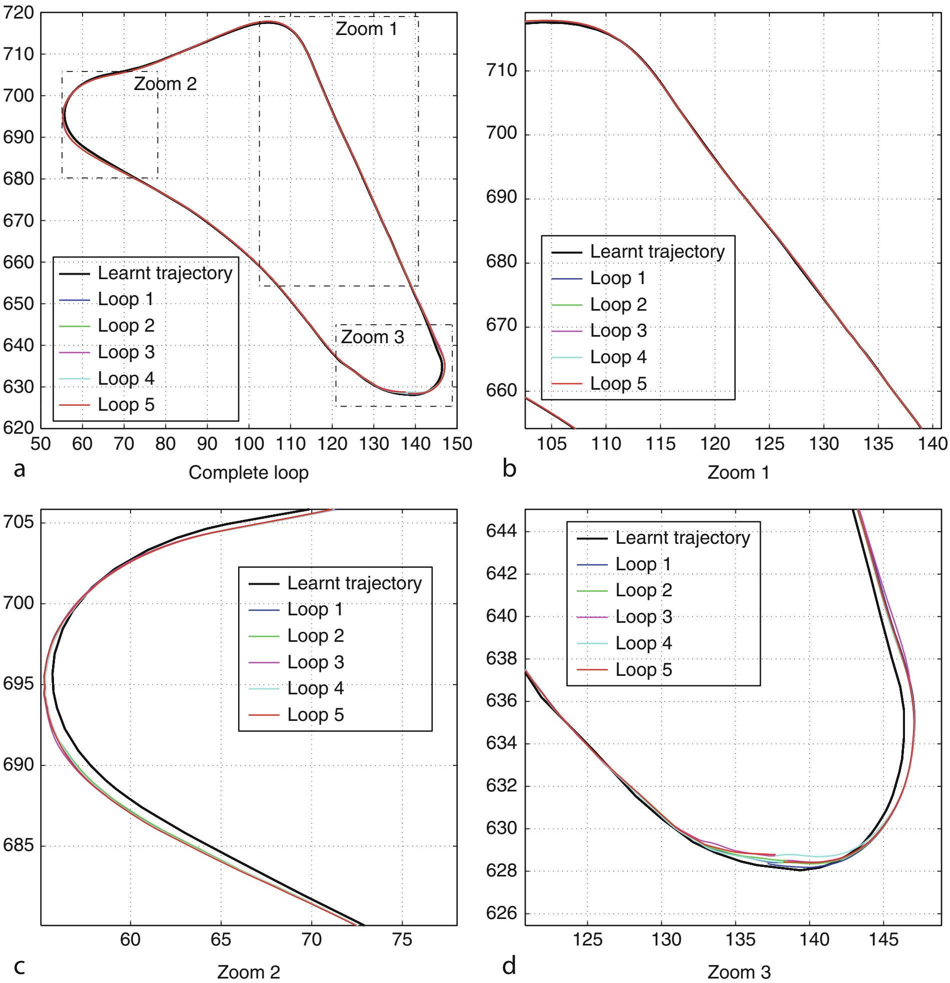


Fig. 53.8

The test loop: Trajectory followed during the learning and the autonomous stages

5.3 Large Displacement

This section presents a complete run from path learning to autonomous navigation.

1. *Learning step*: In a second learning phase, the RobuCab was manually driven along the 800-m-long path drawn in blue in ➲ *Fig. 53.11*. This path contains important turns as well as way down and up and a comeback. After the selection step, 800 key images are kept and form the visual memory of the vehicle. Some of those images are represented in ➲ *Fig. 53.9*.

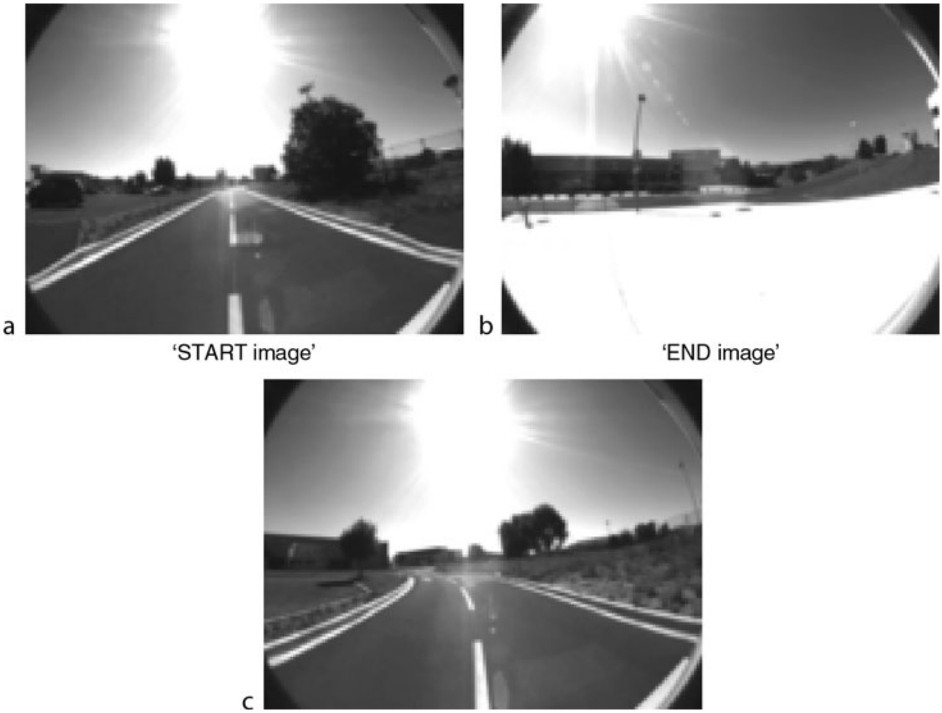


Fig. 53.9
Some key images of the memory

2. *Initial localization*: The navigation task has been started near the visual route to follow (the corresponding image is shown in *Fig. 53.10a*). In this configuration, 15 images of the visual memory have been used in the first stage of the localization process. The distances between the global descriptor of the current image and the descriptor of the memorized images (computed offline) are obtained using ZNCC (*Fig. 53.10b*). After the second step of the localization process, the image shown in *Fig. 53.10c* is chosen as the closest to the image ten (a). Given a goal image, a visual route starting from $\mathcal{I}_i^*$ and composed of 750 key images has been extracted from the visual memory.
3. *Autonomous navigation*: The control (53.3) is used to drive the vehicle along the visual route. A key image is assumed to be reached if the "image error" is smaller than a fixed threshold. In the experiments, the "image error" has been defined as the longest distance (expressed in pixels) between an image point and its position in the desired key image. The longitudinal velocity V is fixed between 1 and 0.4 ms^{-1}. K_p and K_d have been set so that the error presents a double pole located at value 0.3. The vehicle successfully follows the learnt path (refer to *Fig. 53.11*). The experiment lasts 13 min for a path of 754 m. A mean of 123 robust matches for each frame has been found. The mean computational time during the online navigation was of 82 ms by image. As can be observed in *Fig. 53.12*, the errors in the images decrease to zero until reaching a key image. Lateral and angular errors as well as control input are

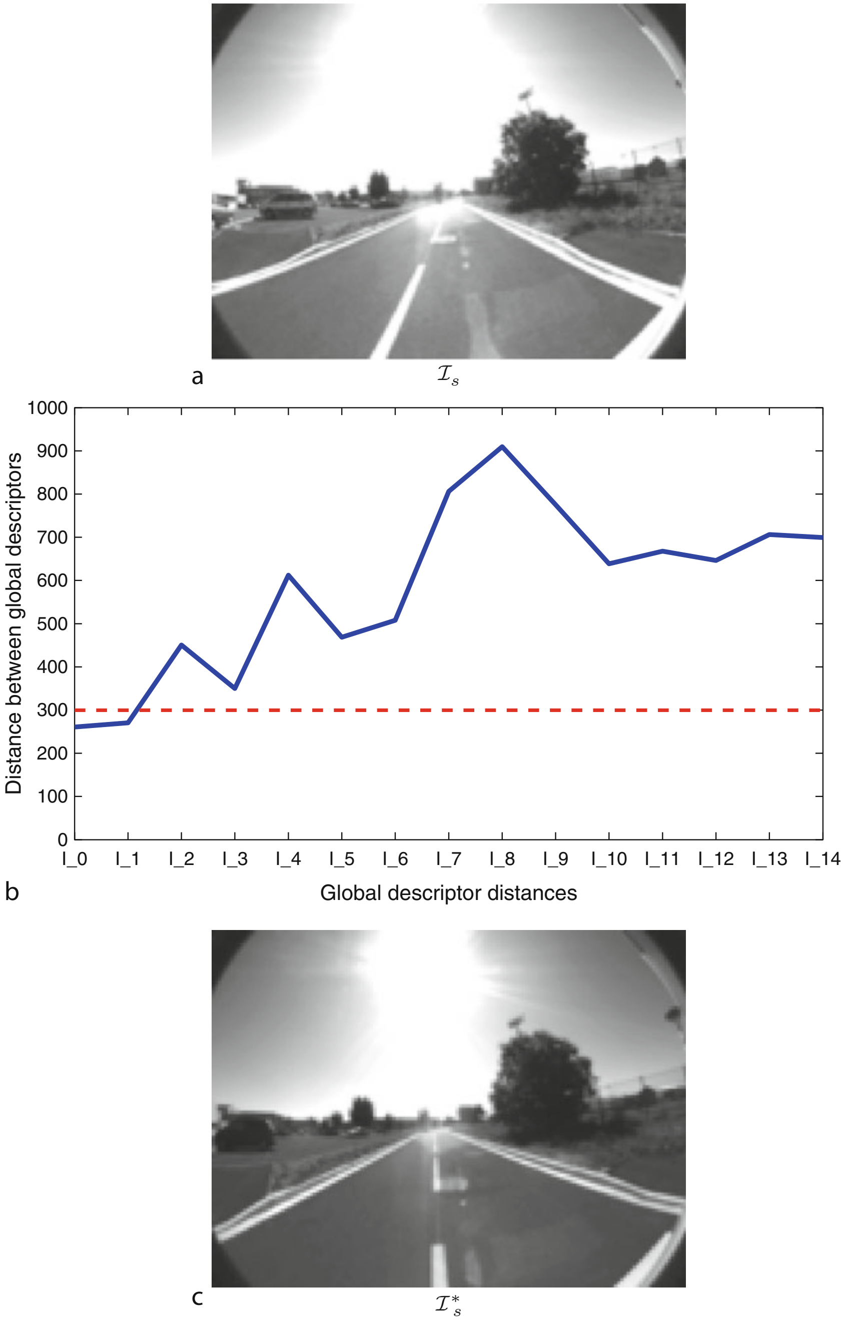


Fig. 53.10
Localization step: $\mathcal{I}_s$ is the current initial image. The distance between the current initial image and the key images global descriptors is drawn in (b). After using the local descriptors, $\mathcal{I}_s^*$ is selected as the correct image

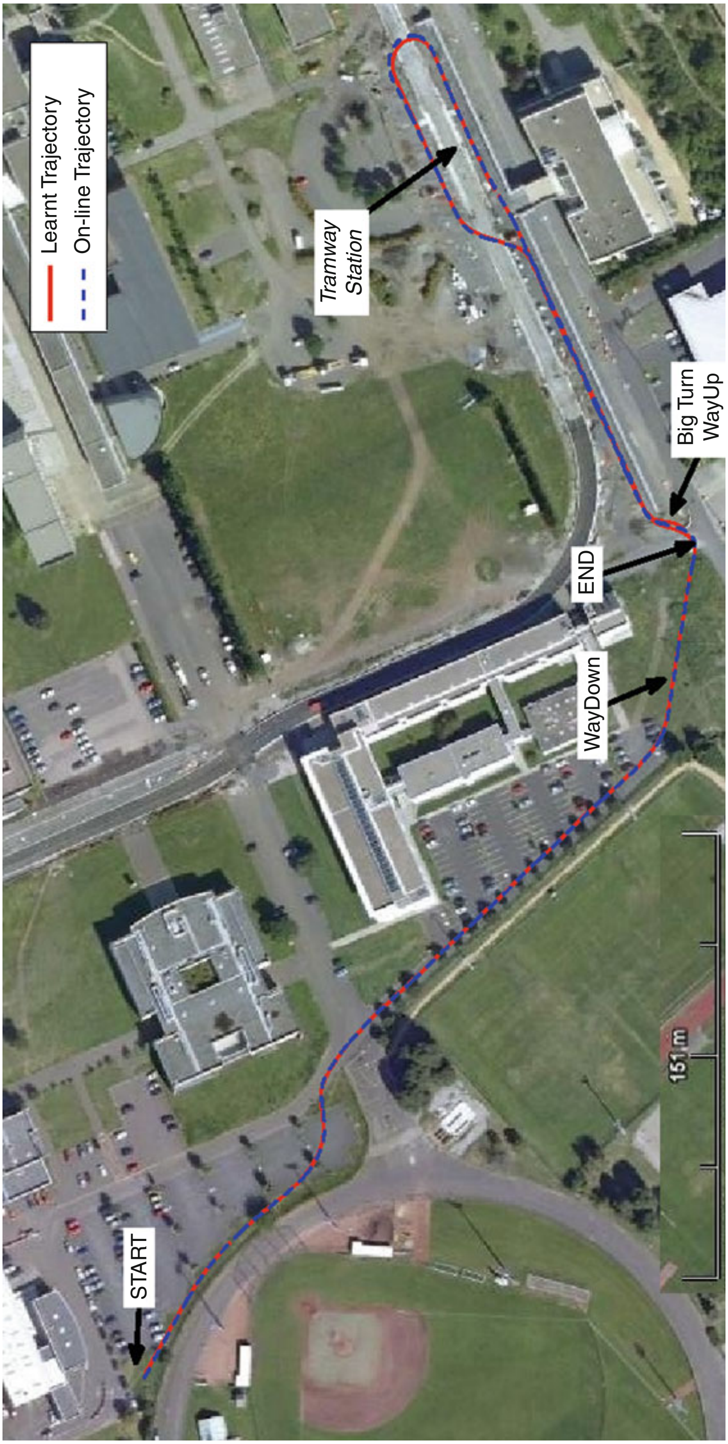


Fig. 53.11

Paths in the university campus executed during the memorization step (in *gray*) and the autonomous step (in *black*)

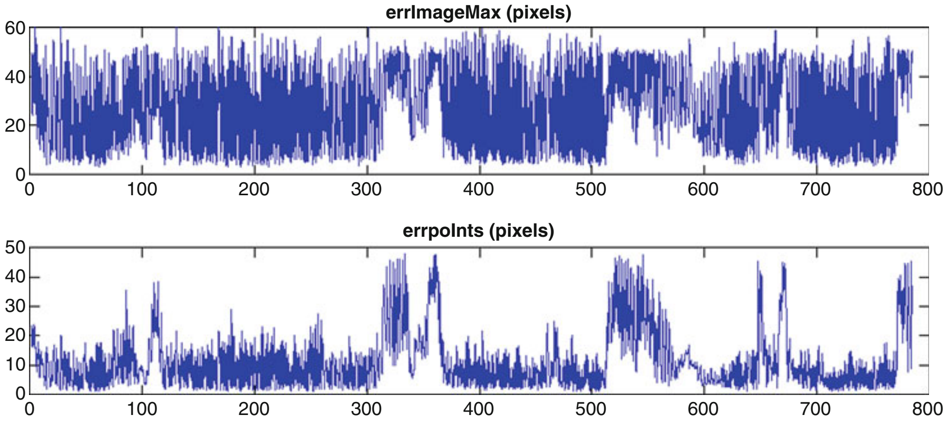


Fig. 53.12

Errors in the images (pixels) versus time (second)

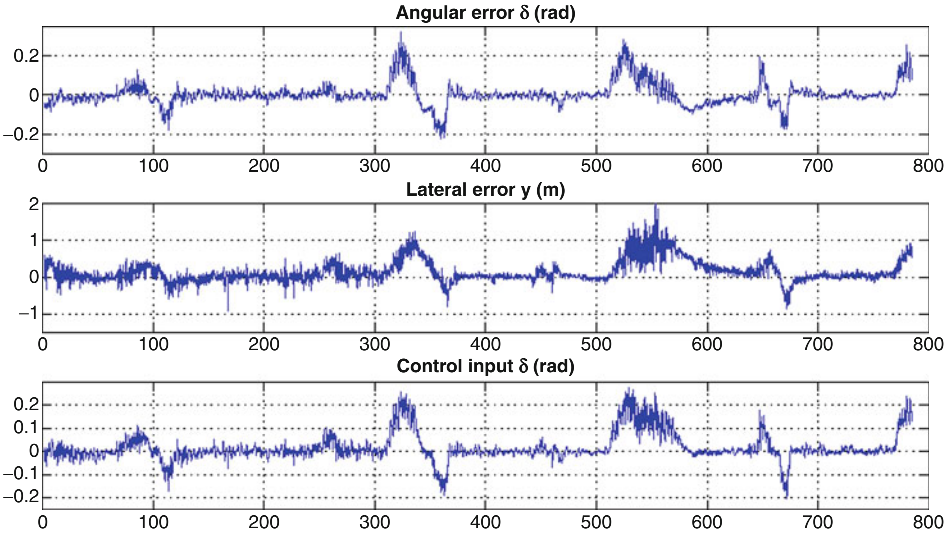
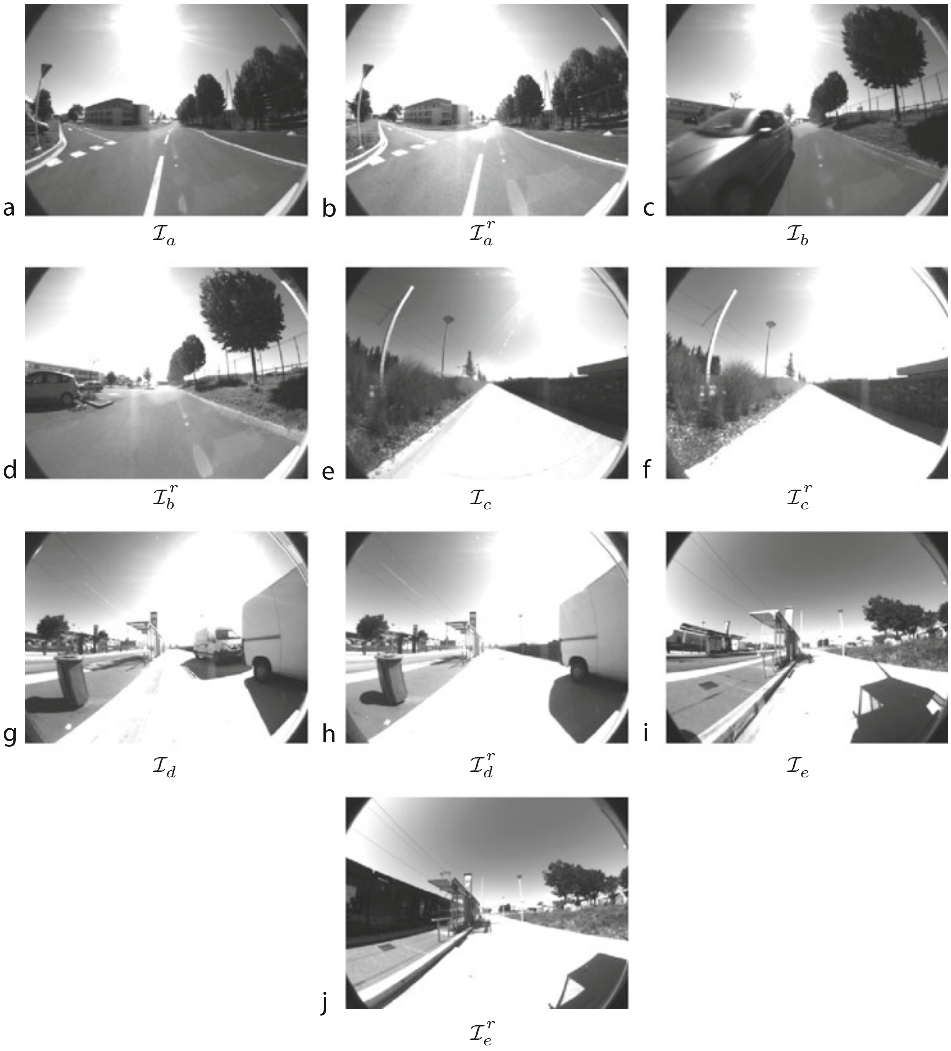


Fig. 53.13

Lateral *y* and angular *θ* errors and control input *δ* vs time

represented in *Fig. 53.13*. As it can be noticed, those errors are well regulated to zero for each key view. Discontinuities due to transitions between two successive key images can also be observed in *Fig. 53.13*.

Some reached images (with the corresponding images of the memory) are shown in *Fig. 53.14*. Note that illumination conditions have changed between the memorization and the autonomous steps (refer to *Fig. 53.14a* and *b* for example) as well as the contents (refer to *Fig. 53.14i* and *j* where a tram masks many visual features during the autonomous navigation).



a $\mathcal{I}_a$

b $\mathcal{I}_a^r$

c $\mathcal{I}_b$

d $\mathcal{I}_b^r$

e $\mathcal{I}_c$

f $\mathcal{I}_c^r$

g $\mathcal{I}_d$

h $\mathcal{I}_d^r$

i $\mathcal{I}_e$

j $\mathcal{I}_e^r$

Fig. 53.14

Some of the current images $\mathcal{I}_k^r$ where the key images $\mathcal{I}_k$ have been reached

4. *Evaluation with a RTKGPS*: The experimental vehicle has been equipped with a Real-Time Kinematic Differential GPS (Thales Sagitta model). It is accurate to 1 cm (standard deviation) in a horizontal plane when enough satellites are available. The accuracy on a vertical axis is only 20 cm on the hardware platform. The vertical readings are thus discarded and the reported errors are measured in an horizontal plane.

DGPS data have been recorded during the learning and the autonomous stages. The results are reported in *Fig. 53.15*. The red and blue plain lines represent

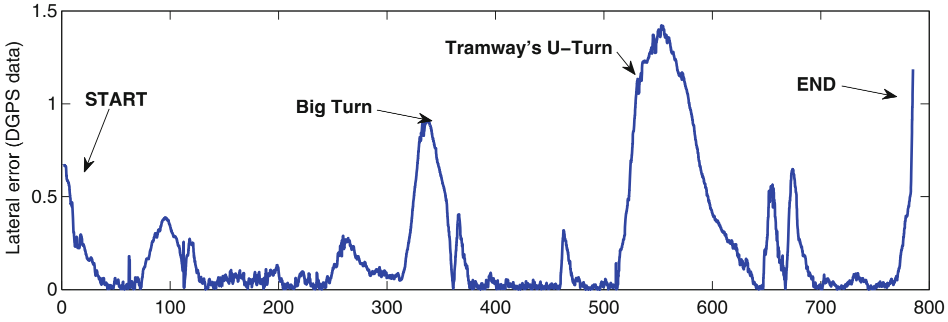


Fig. 53.15

Lateral error (distance between the autonomous and the learnt trajectories, expressed in meter) obtained from DGPS data, vs time

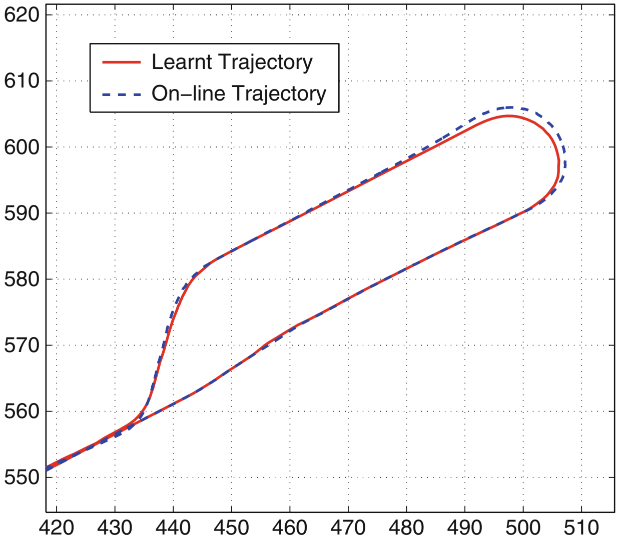


Fig. 53.16

Zoom on the trajectories around the tramway station (positions are expressed in m.)

respectively the trajectories recorded during the learning and autonomous stages. It can be observed that these trajectories are similar.

Distances (lateral error) between the vehicle positions during the learning and autonomous stages are reported on ➋ *Fig. 53.15*. The mean of the lateral error is about 25 cm with a standard deviation of 34 cm. The median error is less than 10 cm. The maximal errors are observed along severe turns (see ➋ *Fig. 53.16* representing a U-turn nearby the tramway station). Note that despite those errors, the visual path is still satisfactory executed (after some images, the vehicle is still at a small distance to the learnt trajectory).

6 Conclusion

This chapter has presented the essential of vision-based topological navigation through an illustrative example. This framework enables a vehicle to follow a visual path obtained during a learning stage using a single camera. The robot environment is modelized as a graph of visual paths, called visual memory from which a visual route connecting the initial and goal images can be extracted. The robotic vehicle can then be driven along the visual route using vision-based control schemes. Importantly, this framework allows loop closure without extra processing.

References

Andreasson H, Treptow A, Duckett T (2005) Localization for mobile robots using panoramic vision, local features and particle filter. In: IEEE international conference on robotics and automation, ICRA'05, Barcelone, Espagne, Apr 2005, pp 3348–3353

Andreasson H, Treptow A, Duckett T (2007) Self-localization in non-stationary environments using omni-directional vision. Robot Auton Syst 55(7):541–551

Atkinson R, Shiffrin R (1968) Human memory: a proposed system and its control processes. In: Spence KW, Spence JT (eds) The psychology of learning and motivation. Academic, New York

Bacca B, Salvi J, Batlle J, Cufi X (2010) Appearance-based mapping and localization using feature stability histograms. Electron Lett 46(16):1120–1121

Bibby C, Reid I (2007) Simultaneous localisation and mapping in dynamic environments (slamide) with reversible data association. In: Robotics: science and systems, Atlanta, GA, USA

Blaer P, Allen P (2002) Topological mobile robot localization using fast vision techniques. In: IEEE international conference on robotics and automation, ICRA'02, Washington, USA, May 2002, pp 1031–1036

Chen J, Dixon WE, Dawson DM, McIntire M (2003) Homography-based visual servo tracking control of a wheeled mobile robot. In: Proceeding of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, Nevada, Oct 2003, pp 1814–1819

Courbon J, Mezouar Y, Martinet P (2008) Indoor navigation of a non-holonomic mobile robot using a visual memory. Auton Robots 25(3):253–266

Courbon J, Mezouar Y, Martinet P (2009) Autonomous navigation of vehicles from a visual memory using a generic camera model. Intell Transport Syst (ITS) 10:392–402

Dayoub F, Duckett T (2008). An adaptative appearance-based map for long-term topological localization of mobile robots. In: IEEE/RSJ international conference on intelligent robots and systems, IROS'08, Nice, France, Sep 2008, pp 3364–3369

Dayoub F, Duckett T, Cielniak G (2010) Short- and long-term adaptation of visual place memories for mobile robots. In: Remembering who we are- human memory for artificial agents symposium, AISB 2010, Leicester, UK

DeSouza GN, Kak AC (2002) Vision for mobile robot navigation: a survey. IEEE Trans Pattern Anal Mach Intell 24(2):237–267

Fang Y, Dawson D, Dixon W, de Queiroz M (2002) Homography-based visual servoing of wheeled mobile robots. In: Conference on decision and control, Las Vegas, NV, Dec 2002, pp 2866–2871

Gaspar J, Winters N, Santos-Victor J (2000) Vision-based navigation and environmental representations with an omnidirectional camera. IEEE Trans Robot Autom 16:890–898

Goedemé T, Tuytelaars T, Vanacker G, Nuttin M, Gool LV, Gool LV (2005) Feature based omnidirectional sparse visual path following. In: IEEE/RSJ international conference on intelligent robots and systems, Edmonton, Canada, Aug 2005, pp 1806–1811

Gonzalez-Barbosa J, Lacroix S (2002) Rover localization in natural environments by indexing panoramic images. In: IEEE international conference on robotics and automation, ICRA'02, vol 2, Washington, DC, USA, May 2002, pp 1365–1370

Harris C, Stephens M (1988) A combined corner and edge detector. In: The fourth alvey vision conference, Manchester, UK, pp 147–151

Hayet J, Lerasle F, Devy M (2002) A visual landmark framework for indoor mobile robot navigation. In: IEEE international conference on robotics and automation, ICRA'02, Washington, DC, USA, May 2002, pp 3942–3947

HochdorferS, Schlegel C (2009) Towards a robust visual SLAM approach: addressing the challenge of life-long operation. In: 14th international conference on advanced robotics, Munich, Germany

Ieng S, Benosman R, Devars J (2003) An efficient dynamic multi-angular feature points matcher for catadioptric views. In: Workshop OmniVis'03, in conjunction with computer vision and pattern recognition (CVPR), vol 07, Wisconsin, USA, Jun 2003, p 75

Jones S, Andresen C, Crowley J (1997) Appearance-based process for visual navigation. In: IEEE/RSJ international conference on intelligent robots and systems, IROS'97, vol 2, Grenoble, France, pp 551–557

Lemaire T, Berger C, Jung I, Lacroix S (2007) Vision-based slam: stereo and monocular approaches. Int J Comput Vision 74(3):343–364

Linåker F, Ishikawa M (2004) Rotation invariant features from omnidirectional camera images using a polar higher-order local autocorrelation feature extractor. In: IEEE/RSJ international conference on intelligent robots and systems, IROS'04, vol 4, Sendai, Japon, Sep 2004, pp 4026–4031

Lowe D (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vision 60 (2):91–110

Ma Y, Kosecka J, Sastry SS (1999) Vision guided navigation for a nonholonomic mobile robot. IEEE Trans Robot Autom 15(3):521–537

Matsumoto Y, Inaba M, Inoue H (1996) Visual navigation using view-sequenced route representation. In: IEEE international conference on robotics and automation, ICRA'96, vol 1, Minneapolis, Minnesota, USA, Apr 1996, pp 83–88

Matsumoto Y, Ikeda K, Inaba M, Inoue H (1999) Visual navigation using omnidirectional view sequence. In: IEEE/RSJ international conference on intelligent robots and systems, IROS'99, vol 1, Kyongju, Corée, Oct 1999, pp 317–322

Menegatti E, Zoccarato M, Pagello E, Ishiguro H (2003) Hierarchical image-based localisation for mobile robots with monte-carlo localisation. In: European conference on mobile robots, ECMR'03, Varsovie, Pologne, Sep 2003

Mouragnon E, Lhuillier M, Dhome M, Dekeyser F, Sayd P (2009) Generic and real-time structure from motion using local bundle adjustment. Image Vision Comput 27(8):1178–1193

Murillo A, Guerrero J, Sagüés C (2007) SURF features for efficient robot localization with omnidirectional images. In: IEEE international conference on robotics and automation, ICRA'07, Rome, Italie, Apr 2007, pp 3901–3907

Nistér D (2004) An efficient solution to the five-point relative pose problem. Trans Pattern Anal Mach Intell 26(6):756–770

Pajdla T, Hlaváč V (1999) Zero phase representation of panoramic images for image based localization. In: 8th international conference on computer analysis of images and patterns, Ljubljana, Slovénie, Sep 1999, pp 550–557

Pollefeys M, Gool LV, Vergauwen M, Verbiest F, Cornelis K, Tops J, Koch R (2004) Visual modeling with a hand-held camera. Int J Comput Vision 59(3):207–232

Royer E, Lhuillier M, Dhome M, Lavest J-M (2007) Monocular vision for mobile robot localization and autonomous navigation. Int J Comput Vision 74:237–260, special joint issue on vision and robotics

Samson C (1995) Control of chained systems. Application to path following and time-varying stabilization of mobile robots. IEEE Trans Autom Control 40(1):64–77

Svoboda T, Pajdla T,(2001) Matching in catadioptric images with appropriate windows and outliers removal. In: 9th international conference on computer analysis of images and patterns, Berlin, Allemagne, Sep 2001, pp 733–740

Tamimi A, Andreasson H, Treptow A, Duckett T, Zell A (2005) Localization of mobile robots with omnidirectional vision using particle filter and iterative SIFT. In: 2nd European conference on mobile robots (ECMR), Ancona, Italie, Sep 2005, pp 2–7

Thormählen T, Broszio H, Weissenfeld A (2004) Keyframe selection for camera motion and structure estimation from multiple views. In: 8th European conference on computer vision (ECCV), Prague, Czech Republic, May 2004, pp 523–535

Thuilot B, Bom J, Marmoiton F, Martinet P (2004) Accurate automatic guidance of an urban electric vehicle relying on a kinematic GPS sensor. In: 5th IFAC symposium on intelligent autonomous vehicles, IAV'04, Instituto Superior Técnico, Lisbonne, Portugal, Jul 2004

Torr P (2002) Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. Int J Comput Vision 50(1):35–61

Triggs B, McLauchlan P, Hartley R, Fitzgibbon A (2000) Bundle adjustment – a modern synthesis. In: Triggs B, Zisserman A, Szeliski R (eds) Vision algorithms: theory and practice, vol 1883, Lecture notes in computer science. Springer, Berlin, pp 298–372

Tsakiris D, Rives P, Samson C (1998) Extending visual servoing techniques to nonholonomic mobile robots. In: GHD Kriegman, A Morse (eds) The confluence of vision and control, LNCIS, vol 237. Springer, London/New York, pp 106–117

Wangsiripitak S, Murray D (2009) Avoiding moving outliers in visual SLAM by tracking moving objects. In: IEEE international conference on robotics and automation, ICRA'09, Kobe, Japan, pp 705–710

Yamauchi B, Langley P (1997) Spatial learning for navigation in dynamic environments. IEEE Trans Syst Man Cybern 26(3):496–505

Zhang Z, Deriche R, Faugeras O, Luong Q-T (1995) A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Artif Intell J 78:87–119

Zodiac T (1995) In: dewit Canedas C, Siciliano B, Bastin G (eds) Theory of robot control. Springer, Berlin