Chapter 12
Autonomous Navigation for Mobile Robots with Human-Robot Interaction

James Ballantyne, Edward Johns, Salman Valibeik, Charence Wong, Guang-Zhong Yang

Abstract Dynamic and complex indoor environments present a challenge for mobile robot navigation. The robot must be able to simultaneously map the environment, which often has repetitive features, whilst keep track of its pose and location. This chapter introduces some of the key considerations for human guided navigation. Rather than letting the robot explore the environment fully autonomously, we consider the use of human guidance for progressively building up the environment map and establishing scene association, learning, as well as navigation and planning. After the guide has taken the robot through the environment and indicated the points of interest via hand gestures, the robot is then able to use the geometric map and scene descriptors captured during the tour to create a high-level plan for subsequent autonomous navigation within the environment. Issues related to gesture recognition, multi-cue integration, tracking, target pursuing, scene association and navigation planning are discussed.

12.1 Introduction

As demands for mobile robots continue to increase, so does the pursuit for intelligent, autonomous navigation. Autonomous navigation requires the robot to understand the environment, whether static or dynamic, and to interact with people seamlessly. In practice, there are several key components that enable a robot to behave intelligently. They include localization and mapping, scene association, human-robot interaction, target pursuing and navigation. Localization and mapping is a well studied topic in robotics and autonomous vehicles for dealing with both known and

James Ballantyne, Edward Johns
Institute of Biomedical Engineering, Imperial College of London, UK e-mail: {james.ballantyne;edward.johns09;salman.valibeik05; charence.wong05}@imperial.ac.uk

Guang-Zhong Yang
Institute of Biomedical Engineering, Imperial College of London, UK e-mail: gzy@doc.ic.ac.uk

unknown environment whilst keeping track of the current location. For purposeful navigation, it also requires learning and scene association to build progressively the surrounding environment. For complex scenes, such as those encountered in a crowded indoor setting, gesture recognition is necessary to ensure seamless human-robot interaction so that they can follow specific commands or pursue relevant tasks. Fig. 13.1 outlines an example configuration when these components are required to work together for autonomous navigation within an indoor environment.

In terms of human robot interaction, vision based approaches represent a key technique for establishing natural and seamless interaction. For understanding human gesture or intention, static or dynamic hand gestures and facial expression can be used [10, 15, 24, 29, 42]. Static gesture normally relies on identifying different postures whereas dynamic gestures include interpreting cascade of events through different time space. In other words, static gestures are extracted by analyzing the contextual information at each time instance, whereas dynamic gestures are recognized by analyzing the temporal information across consecutive time periods. Effective use of human-robot interaction enables a person to initiate various tasks for the robot to carry out. In this chapter, we will use human guided exploration for a robot in a novel environment as an example. The technical details for gesture recognition are described in Section 13.2. Key to any successful gesture recognition system is the incorporation of natural, socially acceptable gestures similar to those used in human-human interaction. The technical details for gesture recognition are described in Section 13.2.

Following a guide also requires the robot to maintain and keep track of the location of the person continuously. To this end, a tracking system as described in Section 13.3 is proposed. The method is based on the use of multiple cues from two main sensor modalities based on vision and laser scanning systems. The visual cues from each sensor are fused to create a robust map of people within the environment. Once the location of the guide is obtained, the robot is able to follow the guide through the environment whilst avoiding visible obstacles autonomously visible obstacles. The basic approaches used are also described in Section 13.3.Even with human guidance, However, situations may arise when the person goes outside the field-of-view of the robot. In this situation, the robot needs to predict where the guide may end up and autonomously navigate to the position and re-establish visual tracking.

In order to build a global map of the new environment through a guided tour, qualitative localization is necessary. We have proposed in Section 13.4 a concept called scene association, which enables the robot to identify salient features of different locations as it navigates around. This information is then incorporated with the internal map generated at relevant locations. The proposed scene association framework uses visual data to learn key features of a scene, which are distinctive but can be consistently identified from different viewpoints.

After the guide has taken the robot through the environment and indicated the points of interest via hand gestures, the robot is then able to use the geometric map and scene descriptors captured during the tour to create a high-level plan for subsequent autonomous navigation within the environment. In Section 13.5, an A* graph

search algorithm is used to plan the route of the robot for goal directed navigation and localization. We will also discuss how learning techniques can be used to improve the robot's ability for autonomous navigation. Throughout this chapter, the examples used are for indoor environments with people moving around. So the proposed framework is ideally suited for museum, office, home-care and hospital wards. The theoretical concepts of using directed navigation to reinforce vision based autonomous localization and mapping can also be extended to other environment. To this end, human gesture recognition can be replaced by other signalling methods, but the basic concept of scene association and high-level planning can remain the same.

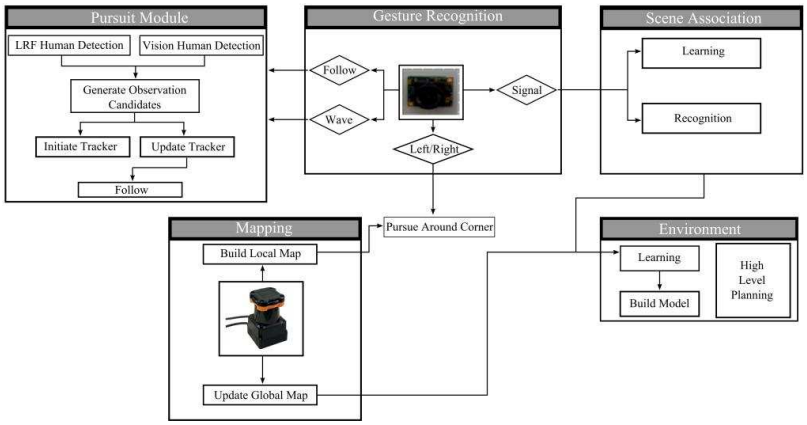


Fig. 12.1 A schematic illustration of the use of gesture recognition, pursuit, scene association and environment mapping for human guided navigation. Under this scheme, the gesture recognition system detects commands issued by a guide, which then activates a specific component based on the detected gesture. The pursuit component is activated on specific gestures and when an "*attention*" gesture is detected, a scene descriptor is built, which is then integrated with the environment model.

12.2 Human-Robot Interaction

In this section, we will describe a robust gesture recognition framework suitable for human guided navigation in normal indoor environment including crowded scenes. For this purpose, the method proposed in [35] is to be used. In this approach, vision based dynamic hand gestures are derived for robotic guidance. The gestures used include "*hello*" (wave gesture for initialization), "*turn left*", "*turn right*", "*follow forward*" and "*attention*" (for building new scene descriptors). The overall system structure is depicted in Fig. 13.2.

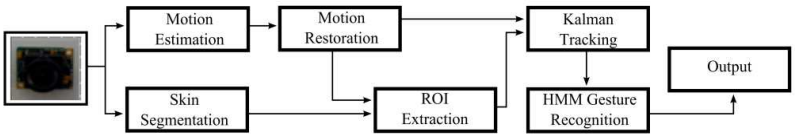


Fig. 12.2 Key processing components for the proposed gesture recognition sub-system. Raw images taken from the vision sensor are used to extract low-level cues such as motion information and skin colored objects. Regions of Interest (ROI) consisting of dynamic skin regions are then extracted and tracked with Kalman Filters. Finally, in the high-level reasoning phase, a Hidden Markov Model (HMM) is used to extract specific gestures

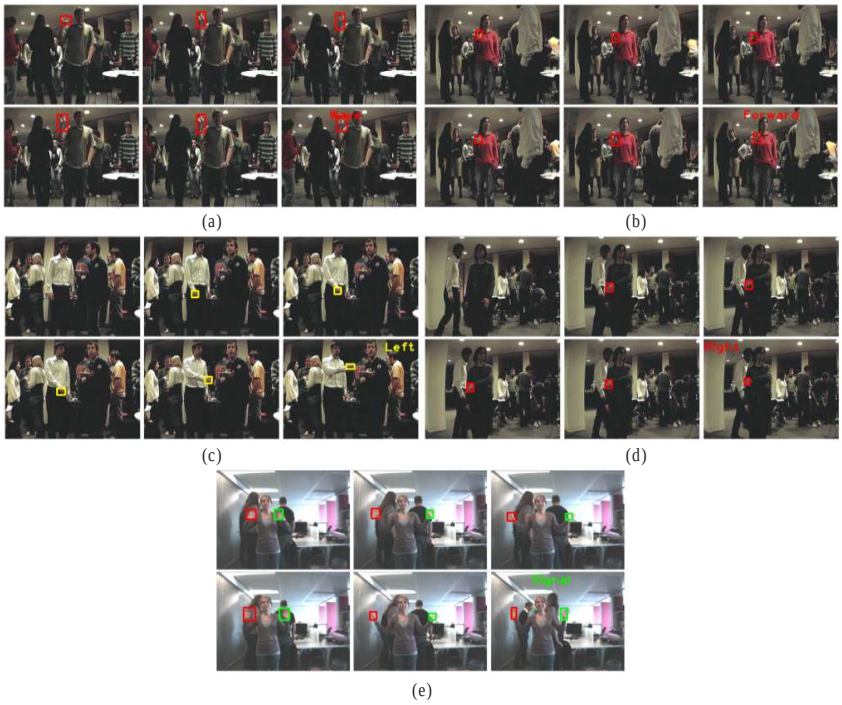
Hitherto, three distinctive factors are commonly employed for extracting hands for gesture recognition. These include skin color, hand motion and shape. We have integrated the first two components, since hand shape is not robust enough for systems using wide angle cameras. In the proposed method, skin segmentation is first performed to identify skin-colored objects which consist of faces, hands or any other skin colored regions. Subsequently, motion segmentation is used to prune out background objects, which are mostly static. The remaining skin-colored objects now mostly consist of hands and faces.

In order to extract temporal information suitable for dynamic gesture recognition, a robust tracking algorithm is required. To this end, Least Median Square Error (LMeds) motion restoration is performed to remove outliers due to rapid illumination changes, partial motion occlusions and depth discontinuities. In practice, object deformation is also important to consider, which is more evident for tracking non-rigid objects. In this work, hand tracking is mainly used to extract dynamic hand gestures. With the proposed framework, pose variability and occlusions are also taken into account by incorporating multiple cues to associate the extracted regions of interest at each time instance to previous measurements. Kalman-filter is used to provide robust tracking across time [35].

Once the hand motion trajectories are accurately tracked, the next step is to perform detailed motion analysis to evaluate if the extracted trajectory is similar to pre-defined gestures. For this purpose, Hidden Markov Models (HMMs) are used. HMM is particularly suitable for modeling time series. The main advantage is that it is based on a probabilistic framework and is beneficial when multiple gestures are evaluated for the same sequence.

To demonstrate the practical value of the proposed gesture recognition framework, Fig. 13.3 demonstrates some example results when different subjects are asked to perform the aforementioned gestures for human guided navigation. Initially, the wave gesture is used to attract the attention of the robot. Subsequently, the robot can be guided by using "*move forward*", "*turn left*" or "*turn right*" commands.

In Fig. 13.3, all the tracked objects are color coded and the recognized gestures are illustrated. After successful gesture recognition, the next challenge is to identify and maintain the commanding person within the field-of-view in a crowded environment. This requires the robot to keep track of the person's location at all times once engaged. Furthermore, the robot must be able to follow the guide through the



(a) (b) (c) (d) (e)

Fig. 12.3 Illustration of different gestures used in the proposed system. Each set of images shows the sequence of motions involved for each gesture. Tracked hand locations are color coded and each detected gesture is indicated in the last image of the sequence; (a), (b), (c), (d) and (e) show "*wave*", "*follow forward*", "*turn "left*", "*turn right*", and "*attention*", respectively.

environment and build a detailed map of the environment for future navigation purposes. In the next section, we will introduce the tracking and pursuit system and explain how the identified gestures are used to control the robot during navigation.

12.3 Subject Following with Target Pursuing

In order to follow the commands given by a person in the scene, the proposed system relies on the robot's ability to detect and track humans based on the sensor data. In this section, we will use information from both vision sensors and Laser Range Finders (LRF) to accurately track and pursue the movement of the person. For human guided navigation, negotiating corners can be problematic as the guiding person can easily move out of the field-of-view. To overcome this problem, the guide can issue a "*left*" or "*right*" signal to activate an autonomous corner manoeuvre during which path planning and obstacle avoidance is performed autonomously.

12.3.1 Correspondence

The proposed framework relies on multiple cues from different sensors to accurately track the guide. However, the cues in the current setup reside in two different camera reference systems. Therefore, calibration is required to fuse both sets of cues into a common reference frame. This establishes a transformation to allow the projection of a laser range point into the vision space. The system utilizes the method defined in [39] to establish the transformation defined as

$$i \approx K(\Phi \cdot P + \nabla) \tag{12.0}$$

where $i = [u,v]^T$, $P = [x,y,z]^T$, and K are the intrinsic parameters of the camera.

To initiate the calibration procedure, a standard checkerboard calibration pattern as proposed by Zhang [39, 40] is used to calibrate the vision sensor. The aim of the procedure is to take multiple instances of the checkerboard within the view of both vision and LRF sensors. The vision calibration provides both the intrinsic parameters,K , and the extrinsic parameters, (R_i, t_i) with respect to each checkerboard location. Furthermore, the extrinsic parameters provide the normal for each checkerboard grid as

$$N = -R_{i,3}(R_{i,3}^T \cdot t_i) \tag{12.0}$$

where $R_{i,3}$is the third column of the rotation matrix for the i^{th} checkerboard obtained from the extrinsic parameters.

After calibration, the laser points on each checkerboard are collected. These points fall on the xz-plane and can be represented by $P^f = [x,z,1]^T$. Therefore, a point falling on the calibration plane with surface normal must satisfy the plane equation $N \cdot P = -D$. From Eq. 12.3.1, we have

$$N \cdot \Phi^{-1}(P^f - \nabla) = -D \tag{12.0}$$

This can be rewritten as

$$\begin{aligned} &N \cdot HP^f = -D \\ &H = \Phi^{-1} \begin{pmatrix} 1 & 0 \\ 0 & 0 & -\Delta \\ 0 & 1 \end{pmatrix} \end{aligned} \tag{12.0}$$

For each pose of the camera, there exist several linear equations for the unknown parameter H, which can be solved with standard linear least squares algorithms. Once H is determined, the relative orientation and position between the two sensors can be calculated as

$$\begin{aligned} &\Phi_R = [H_1, -H_1 \times H_2, H_2]^T \\ &\Delta = -[H_1, -H_1 \times H_2, H_2]^T H_3 \end{aligned} \tag{12.0}$$

where Φ_R is the Rodrigues representation of the rotation matrix. To further enhance the accuracy of the rotation and translation parameters, a non-linear optimiza-

tion technique can be used [39]. It aims to minimize the Euclidean distance from the laser points to the checkerboard grids by using the following equation:

$$\sum_i \sum_j (N_i \cdot (\Phi^{-1}(P_{ij}^f - \Delta)) + D)^2 \tag{12.0}$$

The final result provides the optimized transformation that allows for accurate projection of the laser data points into the image space.

12.3.2 Multi-cue Integration

One of the most common methods for identifying people in vision is to locate the head in each image [11,12,14,17,21,38,41]. These techniques suffer from three major issues; 1) the assumption of the availability of a priori background information; 2) the requirement of large size silhouettes; 3) the need for a controlled environment in terms of illumination changes. Due to the relative positioning of the LRF, laser scanning systems usually identify humans using leg detection schemes. One common approach is to search for local minima [3, 9] in the scan data. This has shown promising results for relatively simple environments. However, as soon as the environment becomes cluttered, the detection results become unreliable and error prone [33]. A second common approach is to use motion detection to identify humans [8, 18] as people are often the only moving objects in most environments. These methods usually compare the current and previous scans to determine the dynamics objects within the environment. The areas from the current scan, which are not found in the previous scan, are considered as the moving objects. The very nature of the algorithm means that the system is not able to detect stationary persons in the environment.

To overcome these drawbacks, we propose to utilize cues from each sensor for person identification. A person is identified if it is evident from both the vision and laser systems. The vision system uses the head detection approach employed by Viola and Jones [36]. In addition, a cascade of adaptive boosting classifiers is used to quickly prune the background and place more emphasis on potential targets. In the examples shown in this chapter, 32 cascades of classifiers are used to provide accurate localization with minimal number of false positives. Furthermore, about 1,399 heads with different orientation, poses and illumination conditions along with 800 background images have been gathered for training. By using adaptive boosting of Haar-like features, a multi-pose head detection classifier has been created. To increase sensitivity, a Kalman filter based tracking system is employed. Head position is updated using Shi-Tomasi features [30]. The method proposed by Valibeik and Yang [35] is used to measure the correlation between newly detected regions with the tracked ones.

The cues from the laser scanning system are formed using a new approach for human detection [4]. The system aims to identify people by searching for three patterns associated with the presence of a person, which are typically found in laser

scans. These patterns include split leg (LSA), forward straddle (FS), and two legs together (SL) as illustrated in Fig. 13.4.

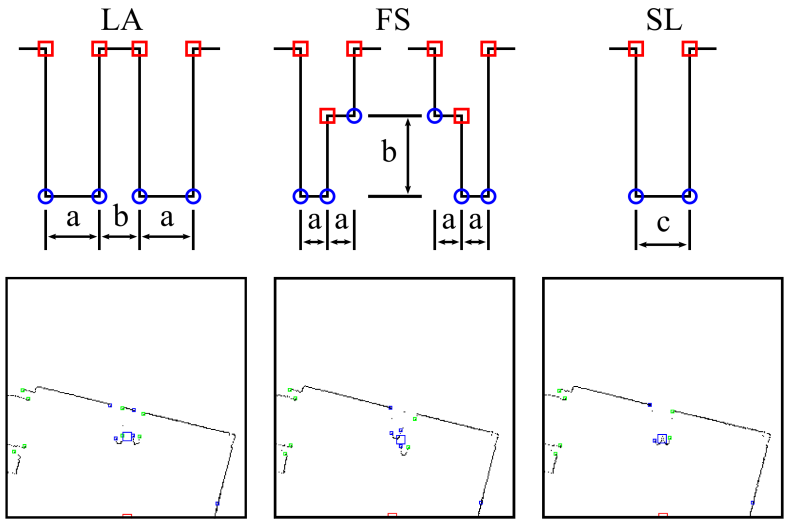


Fig. 12.4 Illustration of the three patterns used to detect human legs in the laser scan data. The top row illustrates the patterns with the three constraints used. In the proposed system, constraint (a) is between 10cm and 40cm, constraint (b) is under 40cm, and constraint (c) is also under 40cm. The bottom row shows a real example from the laser range finder for each type of pattern, where the blue squares represent the right edges, while green squares represent left edges.

The patterns are detected by finding the correct left and right edge sequences where an edge is defined as a segment between two points $\{x_i, x_{i+1}\}$ such that the distance between them is greater than a predefined threshold. An edge is defined as a left edge if $x_i > x_{i+1}$ and a right edge if $x_i < x_{i+1}$.The edges are generated and stored in a list $\sum = \{e_1, e_2, \cdots, e_n\}$. The aim of the algorithm is to find a subset of the edges that follow one of the three patterns with constraints on the size between each segment. The patterns are defined as:

1. LA pattern with quadruplet $\{L, R, L, R\}$.
2. FS pattern with triplet $\{L, L, R\}$ or $\{L, R, R\}$
3. SL pattern with doublet $\{L, R\}$

A single edge from the list can only belong to a single pattern and is thus removed from further consideration. Furthermore, each pattern is searched sequentially, *i.e.*, the edge list is initially search for LA patterns, then for FS patterns, and finally for SL patterns. This is to help reduce the number of false positives.

In order to reduce the number of false positives from each sensor individually, the system fuses both sets of candidates into a single list. To this end, all people detected in the laser scan data are projected into image space using the transformation

found in Eq. 12.3.1. Only those candidates that fall into the horizontal view of the camera are considered. The remaining candidates are then matched with the head detected in the vision system using a nearest-neighbor approach. Therefore, the final list consists of matched pairs from each sensor.

12.3.3 Robust Tracking

The previous section provides a way of identifying humans in the environment based on cues from both vision and LRF sensors. To maintain a continuous estimate of the location of the commanding person, a temporal tracking system is used. Traditional systems have used cues from multiple sensors to identify targets [2, 13, 19]. The proposed system attempts to handle the tracking problem in a similar fashion by using cues from the two sensor modalities, *i.e.*, vision and laser. As mentioned in Section 13.2, the system is activated when a "*hello*" command is received from a person in the environment. Upon receiving the gesture, the robot identifies the most likely candidate from the observation data set by choosing the head most likely to be part of the arm giving the gesture. This observation is used to initialize the tracking system. For tracking, an Interacting Multiple Model (IMM) [28] filter equipped with three motion models is used to deal with unpredictable movement of people. The IMM filter has been shown to provide more accurate tracking results than using a Kalman filter on its own [28]. The three motion models used assume constant acceleration, constant velocity, or a stationary motion model. The system tracks the location of the guide on the *xz*-plane with a weighted model to provide the most likely estimate of the location of the guide. The key component for ensuring accurate tracking is data association. Potential observations come from the fused information obtained from the two sensors as described in Section 12.3.2. To help limit the number of potential observations, the minimum gate of the three motion models [7] is used which is defined using a distance metric:

$$d = \sqrt{(y_m - z_i)^T \sum_m^{-1} (y_m - z_i)} \tag{12.0}$$

where $(y_m - z_i)$ is the measurement residual vector and $\sum_m^{-1}$ is the measurement residual covariance matrix. Finally, only observations that fall in the χ^2 distribution with a probability of 99% of the gate are considered. The tracking system ensures that there is always an estimate of the location of the guide, enabling the robot to follow the guide through the environment when requested.

12.3.4 Pursuing

The tracking system provides the robot with the necessary information to follow the commanding person through the environment. In the example presented in this chapter, the robot uses the given position to ensure:

1. Robot is required to face the guide at all times;
2. Robot is required to maintain a distance of roughly 1.5m to the guide;
3. Movements can only be performed if all objects are avoided.

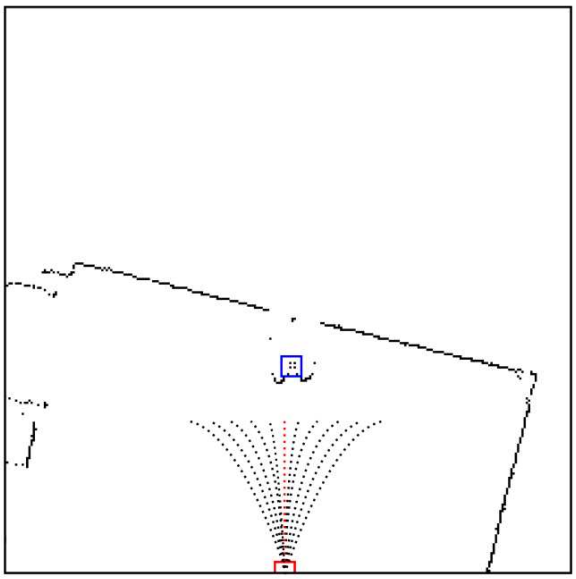


Fig. 12.5 Illustration of the pursuit strategy employed by the robot for following the guide through an indoor scene. The system chooses the best set of velocities that will allow the robot to approach the guide without colliding with any of the objects in the environment. In the above example, the central path is chosen, which is highlighted in red. This path allows the robot to arrive at the desired location at about 1.5m from the guide.

To adhere to these three goals, the robot follows the pursuit movement as defined in [25]. The steering behavior for each frame determines the necessary velocity vector (rotation and translation) that the robot should follow to adhere to Rules (1) and (2). The velocity vector is determined by the current predicated location of the guide and the robot's current velocities.

$$\begin{aligned} v_{desired} &= norm(pos_r - pos_t) \cdot v_{max} \\ v_{actual} &= v_{desired} - v_{current} \end{aligned} \tag{12.0}$$

Upon arriving at the actual velocity to use, the robot must ensure that Rule (3) is preserved. To this end, the robot selects a series of velocities within a window of v_{actual} and generates the curves that the robot would follow at the selected velocities. The robot then chooses the velocity that allows the robot to arrive at the desired location, 1.5m from the guide while avoiding all obstacles. Fig. 13.5 illustrates the potential paths that the robot could take to approach the guide marked with a blue square. In this example, the red, dotted path is chosen because it brings the robot closest to the desired position of 1.5m in front of the guide.

The secondary pursuit goal is to handle situations when the guide leaves the field-of-view of the robot. This situation arises when the guide either goes around a corner or enters into a room through a doorway. To circumvent these problems, the guide is able to direct the robot either with a "*left*" or "*right*" signal depending on the traveling direction. When the robot receives one of the two gestures, it predicts the future location of the guide around the corner in the desired direction. To do this, the robot projects the current location of the guide to the left or right of the field-of-view.

When the projected location has been found, the robot plans a path to the location that avoids all obstacles. An example situation is shown in Fig. 13.6 when the user is leaving a room and moving to the right and down the hallway. The robot chooses a position about 1m to the right of the latest predicted location of the guide, creating a path through the doorway to the goal location shown in red.

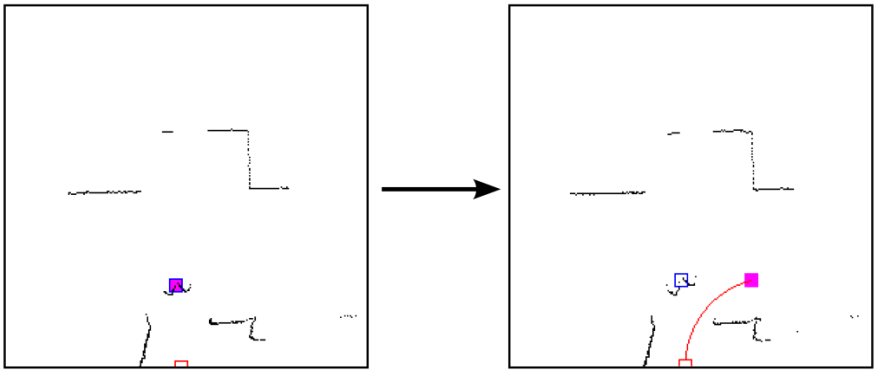


Fig. 12.6 Illustration of the method used by the robot to turn around a corner. The left image presents a guide standing in a hallway, while the robot is still in the room. After the guide has issued a "*turn right*" gesture, the robot assumes the guide will leave the field-of-view and begins to perform a corner maneuver autonomously. The right image illustrates the path in red, generated to allow the robot to arrive at the projected future location of the guide.

12.3.5 Mapping

During a guided tour of an environment, the robot has the ability to use the laser data to build a geometric estimate of the environment. To this end, an occupancy grid is constructed, while the location of the robot in the environment is tracked using a scan-matching system. In the proposed framework, an implementation based on the "*vasco*" scan-matching system, which is part of the Carnegie Mellon Robot Navigation Toolkit [23], is used. A sample map generated for our lab at Imperial College London is shown in Fig. 12.7.

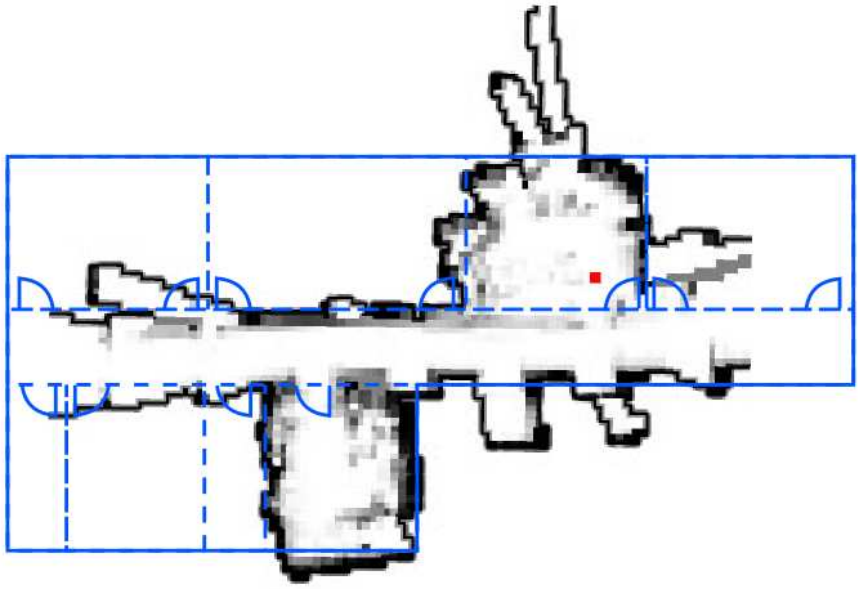


Fig. 12.7 A sample occupancy grid generated during a tour of our lab at Imperial College London. The red square illustrates the current location of the robot in the environment.

The map generated during the tour of the environment provides only a geometric perspective of the environment. This introduces ambiguities as locally, many of the architecture features are very similar across a building. The robot can easily become confused to its actual location. To address this limitation, a vision based scene descriptor is used to build a global perspective based on appearance. These descriptors are generated after receiving a "*signal*" gesture from the guide during the tour.

12.4 Qualitative Localization

The laser mapping system presented so far is capable of calculating the quantitative location of the robot in a defined coordinate system. In practice, however, its robustness is far from perfect, and it is important to provide complementary location information. To this end, our proposed system relies on visual information to qualitatively identify the scene that the robot is currently in view of, working independently of the mapping system, and ultimately with both systems reinforcing each other.

For indoor scenes, rooms are often geometrically similar, and this can cause problems with the proposed mapping system when only building a rough geometrical view of the environment. Visually, however, these rooms are often very distinctive. Features of a room, such as pictures on the wall or lights on the ceiling, present information that a laser system is unresponsive to. Visual information can help reveal the room and significantly reduce the search space for localizing the robot.

The ability of the robot to understand in which room it is located also adds to the pervasive nature of the system. For example, should the robot be required to relay its location for repair, simply stating the name of the room to the engineer is more meaningful than providing a series of numbers representing its location. In addition, visual representations of a scene lend contextual information that laser systems cannot provide. This is of great benefit when the robot is required to interact with its environment.

It is also worth noting that no navigation system should rely entirely upon one type of sensor. Combining visual and laser sensing provides both depth and content information, which presents a sound framework upon which to build a robust navigation system. This overcomes the malfunctioning of a sensor and/or an environment poorly suited to a single sensor.

The compliment to the laser mapping system in the proposed framework is based on scene recognition. Scene recognition for robotic applications is a field that can often be considered as a special case of image matching. Finding the image in a database most similar to a candidate image has been widely addressed in literature. Many approaches represent images by a distribution of features such as SIFT [20], with matches between features based upon similarity in feature descriptors, as well as the spatial relationships of features [26, 27, 31]. The transfer of these techniques to robot localization must deal with the problems associated with indoor scenes. Such scenes often have a lower presence of discriminating features, and instead contain a large number of uniform regions representing commonly-occurring bodies such as walls, floors and ceilings. This results in images not only having fewer features to match, but those features found are often present in other similar rooms. A further issue is that viewpoint changes in indoor environments are often large relative to images of outdoor scenes, which is generally the focus of the above techniques. As such, most approaches for indoor scenes use more advanced methods such as supervised learning [34], probabilistic matching [16], feature grouping [1], or a combination of both global and local features [37].

12.4.1 Scene Association

The ability to recognize specific features of a scene is important for a robot to interact with and navigate within a scene. For this purpose, our method of scene association allows a scene to be recognized from a number of viewpoints, whilst still identifying specific features. We propose that we call features that are viewpoint-invariant and are consistently detected across different viewpoints as *association features*.

In order to extract the association features from a scene, several images of the scene are captured from varying viewpoints, and features which occur across all images are retained. In this work, SIFT features are used. During the training phase, a match is tested between each feature in an image, and all features in the other images of the scene. Those features which are found in all images are retained as association features. In our equations, f_a represents an association feature and f_c represents a candidate feature which we are attempting to match to an association feature. To determine whether a match is made, three steps are taken, and steps with the least computational expense and most likely to eliminate the greatest number of false matches, are handled first.

In the examples shown in this chapter, it is assumed that the robot maintains an upright position, such that the features will only vary by small amounts due to affine viewpoint changes and not absolute camera rotations. Thus a candidate feature is firstly discarded if its orientation differs to that of an association feature by more than a threshold, t_θ:

$$abs(f_a(\theta) - f_c(\theta)) > t_\theta \tag{12.0}$$

Then, for any candidate feature that is not eliminated by (9), the difference in descriptors between f_a and f_c is calculated, by summing the dimension-by-dimension differences between the SIFT descriptors, $d_1 \cdots d_{128}$. The feature is discarded if this difference is more than t_{sift}:

$$\sum_{i=1}^{128} abs(f_a(d_i) - f_c(d_i)) > t_{sift} \tag{12.0}$$

For those candidate features not eliminated by (10), elementary graph theory is then used to eliminate matches that are not verified by the local neighborhood. The neighborhood of a feature is defined as the 10 spatially-closest features captured in the same image, as proposed in [26]. Then, a feature $f_c^{(n)}$ in the neighbor-hood of f_c is considered a neighborhood match, if there exists a feature $f_a^{(m)}$ in the neighborhood of f_a, which has a similar orientation and descriptor to $f_c(n)$, as defined in Eqs. (9) and (10). Additionally, the angle between f_c and $f_c^{(n)}$ must differ to the angle between f_a and $f_a^{(m)}$ by no more than t_φ. Then the candidate feature f_c is discarded if the number of neighborhood matches to is f_a less than N:

$$\sum_{n=1}^{10} NumMatches(f_c^{(n)}, f_a) < N \tag{12.0}$$

where

$$NumMatches(f_c^{(n)}, f_a) = \begin{cases} 1 \; if \sum_{m=1}^{10} IsMatch(f_c^{(n)}, f_a^{(m)}) \geq 1 \\ 0 \qquad\qquad otherwise \end{cases} \tag{12.0}$$

where

$$IsMatch(f_c^{(n)}, f_a^{(m)}) = \begin{cases} 1 \;\; if \quad abs(f_c^{(n)}(\theta) - f_a^{(m)}(\theta)) < t_\theta \\ \quad and \;\; \sum_{i=1}^{128} abs(f_c^{(n)}(d_i) - f_a^{(m)}(d_i)) < t_{sift} \\ \quad and \; abs(\varphi(f_c^{(n)}, f_c) - \varphi(f_a^{(m)}, f_a)) < t_\varphi \\ 0 \qquad\qquad otherwise \end{cases} \tag{12.0}$$

In the above equation, $\varphi(f_1, f_2)$ represents the orientation of the line connecting features f_1 and f_2. If a candidate feature satisfies all these criteria, then it is considered a match between the two images. It is then passed on to the next image of the scene to determine whether the same feature is found again. Once an association feature is found across all images, its descriptor is calculated by computing the dimension-by-dimension average of the descriptors of all the features contributing to this association feature.

In the example shown below in Fig. 13.9, three images of each scene are used, and an association feature is recorded if it is present in all three images. Using more images can significantly reduce the number of detected association features, thus affecting its ability to perform scene association on a captured image in later stages. The top row shows all the originally detected features, and the bottom row showing only the association features, which were found in all three of the top row images.

In Fig. 13.9, it is worth noting that the association features all form part of the background of the images, and all features on foreground objects are eliminated. There are two reasons for this. First, a background feature across all three images has a similar incident viewpoint than foreground features, and hence the feature descriptor varies less between the viewpoints. Second, background features which lie against a wall have no background clutter to confuse the feature descriptor, whereas the descriptor for foreground features varies as different elements of the background come into view behind the feature.

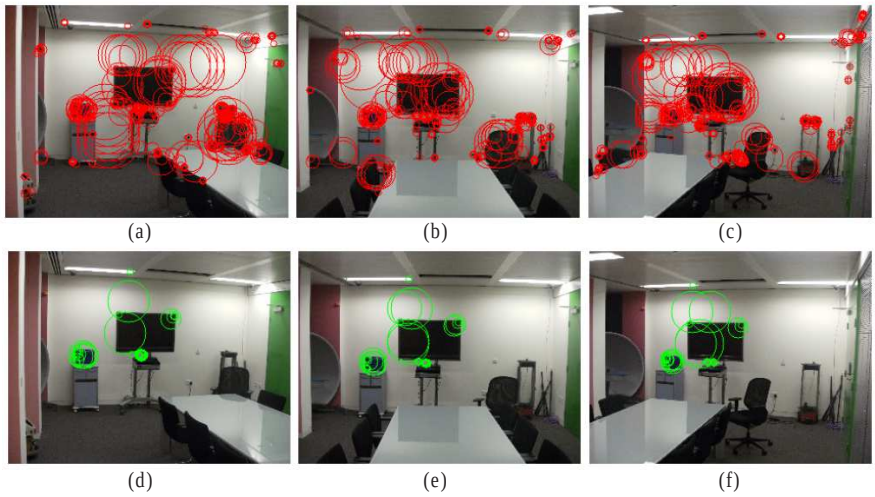


Fig. 12.8 Example of the training phase during which association features are detected for a scene. Images (a) - (c) are taken from different viewpoints of the same scene. SIFT features are then detected in the images and highlighted. Those features that are found in all three images (a) - (c) are memorized as association features and highlighted in images (d) - (f).

12.4.2 Scene Recognition

With association features learnt for each room, the next task is to match features from a captured image as the robot navigates through the environment, to those association features stored in the robot's memory. This is done in a similar manner as during the training phase. First, candidate SIFT features, f_c, are extracted from the latest captured image. Then, for every association feature, f_a , in each scene in memory, a match is attempted to every candidate feature, f_c. A match is classified as positive if it is similar in orientation to f_c, has a similar descriptor to f_c, and is verified by the local neighborhood of f_c. This is identical to the process of learning association features in 4.1, except that we now use a smaller value for t_{sift}. This adjustment is necessary because in the training phase, features are only compared to those from a small number of images of the same scene. However, during the recognition stage, features are compared to features from all scenes in the database, and so descriptors are required to be closer to have sufficient confidence of a match.

Choosing the actual values of t_{sift} in the two phases is a compromise between feature discrimination, and viewpoint invariance. In our example results, we found that for the training phase, $t_{sift} = 25$ was an optimum value, generating a large number of positive matches and leaving only 10% false positive matches, which were then all eliminated during neighborhood verification. For the recognition phase, t_{sift} can be tweaked in accordance with the number of rooms in the environment and for the examples shown in this chapter, $t_{sift} = 45$. With a smaller value, the same feature detected across large viewpoints was often eliminated, and with a larger

value, too many false positive matches were found that could not be eliminated by neighborhood verification.

If a match between an association feature and a candidate feature is positive, the algorithm attempts to find a match to the next association feature. For each scene, the percentage of association features which have been matched then enters the scene into a ranking system, where the scene with the highest percentage of association feature matches is output as the scene within which the robot is located.

Fig. 13.10 demonstrates a typical arrangement within the boundaries of a room where the proposed scene association is used. At each location, the robot captures a series of 8 images at 45° intervals to form a panoramic sequence, and computes SIFT features for each image. Fig. 13.11 shows the panoramic images with all detected features highlighted. An image matched which match to an association feature in memory increases the likelihood of it being associated that scene.

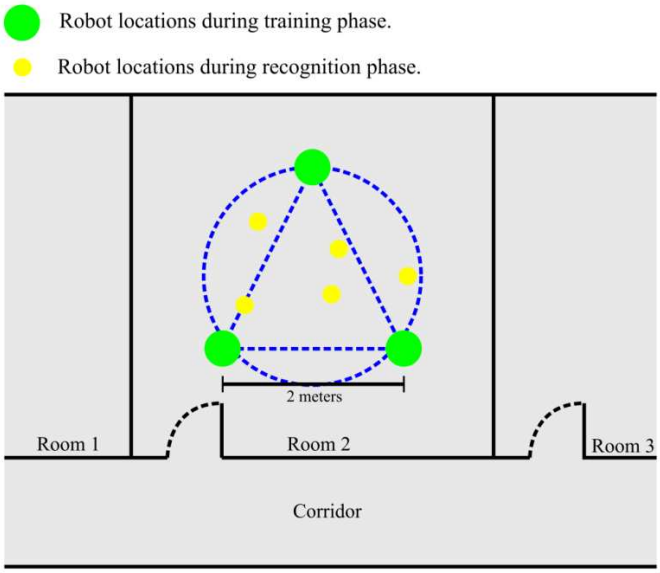


Fig. 12.9 Arrangement of robot locations within a room during training and recognition phases. During the training phase, scenes are captured at three points of the triangle, whereas during the recognition phase, scenes are captured randomly within the circle tangential to the triangle. At each location, the robot rotates to capture multiple images to form a panoramic sequence.

During the training phase, the robot is initially instructed by hand gestures to capture panoramic images in 7 rooms of the building. In each room, the robot learns the association features by capturing images at each of the three locations in Fig. 13.10.

During the recognition stage, the robot captures one set of panoramic views and calculates the percentage of matches to association features for each scene. In this experiment, 93% of the test scenes were identified with the correct room, by consid-

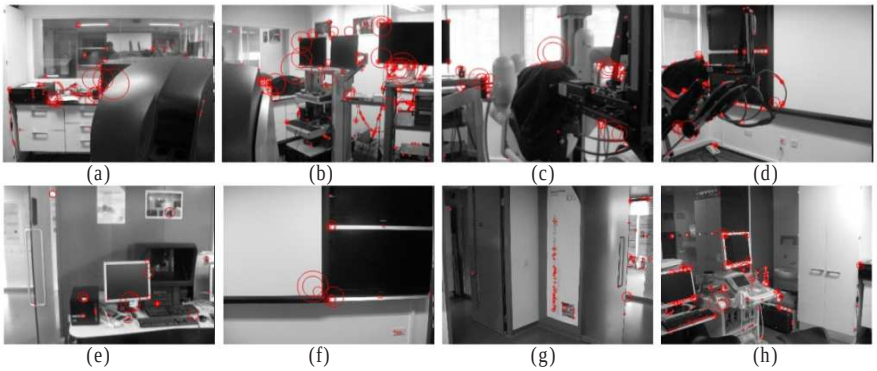


Fig. 12.10 Panoramic images with SIFT features highlighted for scene association. Images (a) - (h) are captured at 45° intervals as the robot rotates within a room. This is performed in both the training and recognition phases.

ering the highest percentage features matches across all scenes in the database. Table 12.1 shows the recognition performance across the seven rooms visited, showing the average results across multiple recognition attempts for each room. The numbers in bold represent the percentage of association features recognized in the correct room (true positives), whilst the non-bold numbers represent the percentage of association features recognized in all the other incorrect rooms (false positives).

Table 12.1 Recognition accuracy by using the scene association method proposed for a laboratory scene consists of 7 rooms. Bold numbers are percentage of true positive feature matches, non-bold numbers are the percentage of false positive feature matches. Parameters used in equations (9) - (11): $t_\theta = 20$, $t_\varphi = 45$, $N = 1$, $t_{sift} = 45$ for training phase, 25 for recognition phase.

	% Feature matches in each room						
Room number	1	2	3	4	5	6	7
1	**60**	4	17	1	9	17	29
2	2	**48**	2	1	0	0	4
3	13	11	**82**	16	27	16	1
4	2	45	0	**55**	42	17	3
5	8	7	6	37	**72**	25	1
6	7	18	3	14	31	**36**	0
7	0	10	7	4	7	3	**63**

It is evident that some rooms have generated a higher confidence in their correct identification. For example, Rooms 1, 2, 3, 5 and 7 have large differences between the most likely and second most likely rooms, whereas with Rooms 4 and 6, the system is less confidence that the most likely room was indeed identified correctly. This is largely due to the presence of similar objects in different rooms, such as

television screens, whose features are similar across different scenes, and who also have similar features in the local neighborhood, drawn from the same object.

Nonetheless, with a 93% positive scene identification, this vision system is well equipped to work in tandem with the laser mapping system, and integrates appropriately with the gesture-recognition task. The final challenge is then to incorporate both the qualitative and quantitative localization data, into a system that is able to autonomously navigate between rooms, as instructed by the user.

12.5 Planning and Navigation

As the robot is guided around the environment, laser data is collected in order to build a geometric map of its surroundings. As mentioned earlier, the guide indicates points of interest within the environment by performing an "*attention*" gesture. The tour is to enable the robot to map the environment using quantitative and qualitative localization techniques; incorporating scene association improves localization and also the high-level planning used for navigation.

After mapping and localization, in order to autonomously navigate towards a goal, there needs to be a plan. A plan can be described as a sequence of moves or reactions which lead towards the goal [22]. Formulating a plan when the environment map is discrete is simpler since classical graph-searching algorithms such as A* and Dijkstra can be used [22]. The two main approaches for discetizing the environment is to either store it as a grid, grid-based (metric) paradigm, or as a graph, topological paradigm [5]. By using the laser mapping system and scene association descriptors, we can integrate both grid-based and topological paradigms to allow for fast path planning on the easy to construct occupancy map, utilizing the advantages of each representation, as mentioned by Thrun and Bücken [5].

During a guided tour, the robot constructs the occupancy map of its environment and, when gestured by the user, records a scene descriptor for its current location, which is mapped onto the occupancy map as shown in Fig. 13.12. In addition to the scene descriptors created, a key location is also indicated. It would also be useful, for navigation purposes, if descriptors are captured automatically through-out the tour as waypoints, since this will allow the topology of the environment to be captured more accurately.

Scene descriptors are periodically captured during the tour, allowing the graph-based map to also contain information about the path taken by the guide, and not just the points of interest; we captured these waypoints when turns greater than 5ąã were made to ensure the robot would be able to later retrace the path taken during automated runs as shown in Fig. 13.13.

Once the robot localizes itself on the occupancy map, we can plan a route to the target locations using the topological map, starting from the current nearest node. This high level planning procedure is done by using the A* graph-search algorithm. The system uses cues from the LRF and camera to recognize when it reaches waypoints or the goal location.

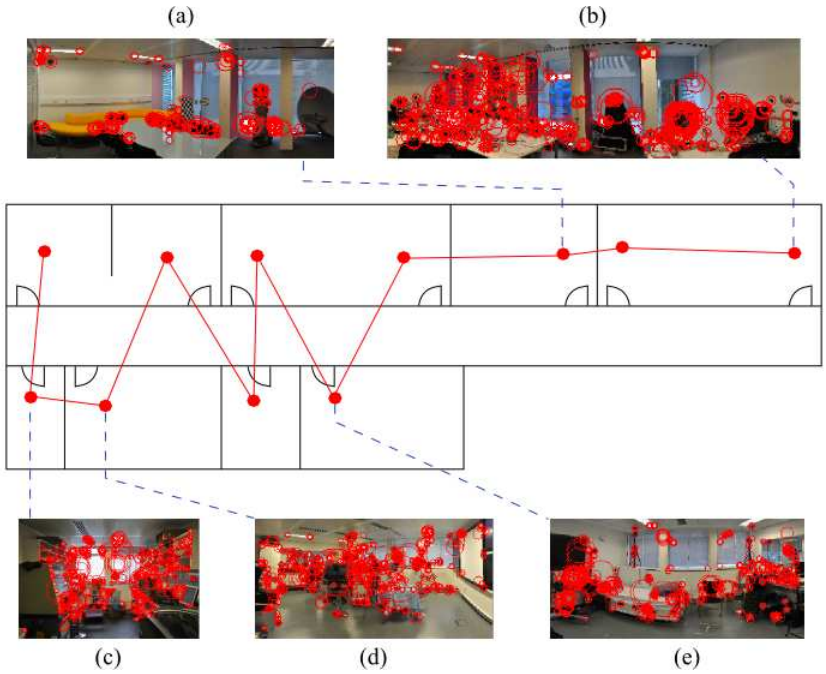


Fig. 12.11 A topological map of the environment storing scene descriptors at the key locations indicated by the user during the tour.

In autonomous systems, learning can potentially provide the flexibility the system needs to adapt to dynamic environments [6]. Consider, for example, that a new optimal path is discovered between two locations, it would be desirable for the robot to update its internal model to reflect this discovery. Based on Thrun's idea of sensor interpretation [32], a learning method which interprets readings from different sensors, such as the laser range finder and color camera, could be utilized for coping with varying environments. For example, in repetitive scenes, such as the corridor shown in Fig. 13.14, the ability for accurate localization using scene recognition would decline dramatically. In such scenarios, it would perhaps be more beneficial if the robot could learn to rely more on other sensory information.

Other factors, besides a changing environment, would benefit from updates to the robot's internal model. Graph searching can be a computationally demanding, especially in complex environments. Our focus is to capture scenes to store as a node on the graph-based map automatically when a significant rotational motion is executed or if a large distance has been covered since the last recorded node. Although the topological map allows for faster planning, when compared to the grid-based occupancy map, the robot should seek to further simplify its representation of the environment as shown Fig. 12.14.

The simplified representation of the environment allows the robot to carry out future tasks in an autonomous fashion. Furthermore, the simplified map provides

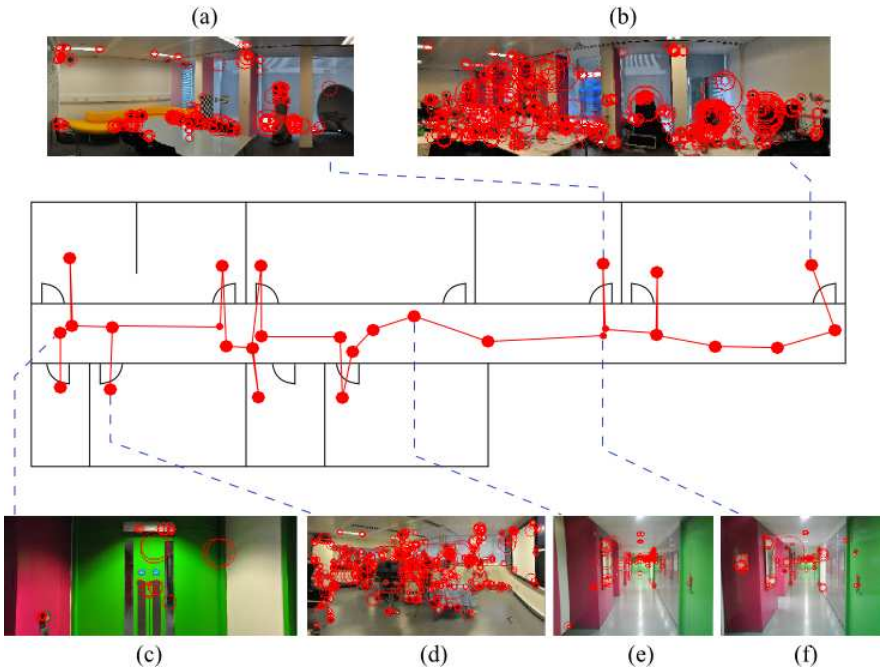


Fig. 12.12 An example topological map built during a guided tour. Capturing scene descriptors periodically during the guided tour allows the robot to build a more detailed topological map of the environment, better recording the path taken by the user.

a user-friendly interface for control of the robot. This type of interface allows the proposed system to work in a variety of environments including museums, offices, home-care and hospital wards. Not only is the robot able to identify different rooms in the environment, whether it be to carry out a task or alert an engineer for repair, but also does the proposed system allow the robot to interact with people in the environment, whilst avoiding all obstacles.

12.6 Conclusion

Mobile robots present many opportunities to carry out mundane tasks in everyday life. Before robots are able to perform such tasks, basic intelligence must be developed. In this chapter, we have addressed several key challenges related to robotic navigation and the value of using HRI for environment mapping and scene association. Effective use of HRI allows the user to naturally interact with a mobile robot via gestures, which can be detected using a vision based system. We have demonstrated the practical use of the proposed gesture recognition system for guided exploration in a novel environment. These gestures help the robot in difficult situations and

(a) (b)

Fig. 12.13 Localization within a scene with repetitive visual features. Scenes (a) and (b) are captured from different locations in the environment, however, many features in both images indicate a match; scene association is not useful in all situations as most of the matches shown are incorrect.

build scene descriptors. Upon being informed to follow, the proposed system used a multi-cue tracking system to maintain an estimate of the location of the guide at all times.

During a guided tour, the robot uses the laser data to create an occupancy map of the environment. However, there are scenarios where localization using this quantitative approach can be improved by using qualitative data. To this end, a vision based scene association is used to complement the occupancy map by capturing descriptors of particular scenes on the map. These descriptors are built at salient locations of the environment. The visual descriptors consist of distributions of SIFT features, which the robot has learned to memorize as they occur consistently across multiple viewpoints of a scene.

To autonomously navigate within the recorded environment, the robot uses both the geometric occupancy map and topological map of the scene descriptors. Quantitative and qualitative localization techniques are complementary with each other, providing accurate localization in geometrically similar environments. To accurately retrace the path taken by the guide, scene information is captured periodically by the robot during the guided tour. High level path planning is carried out by performing A* search on the topological map from the current scene to the goal destination. In this chapter, we have described our considerations on how autonomous navigation can be improved by incorporating mechanisms that will allow it to cope with a changing environment and uncertainty from sensor readings. We demonstrated how visually similar scenes can potentially cause confusion for scene association and suggest how the robot could adapt its interpretation of sensor data under these conditions.

While the proposed system attempts to handle many of the issues related autonomous navigation, future work will aim to improve the robustness of the system. More sensor modalities could be used to further help the robot understand the environment. For example, 3D time-of-flight cameras could be used to accompany

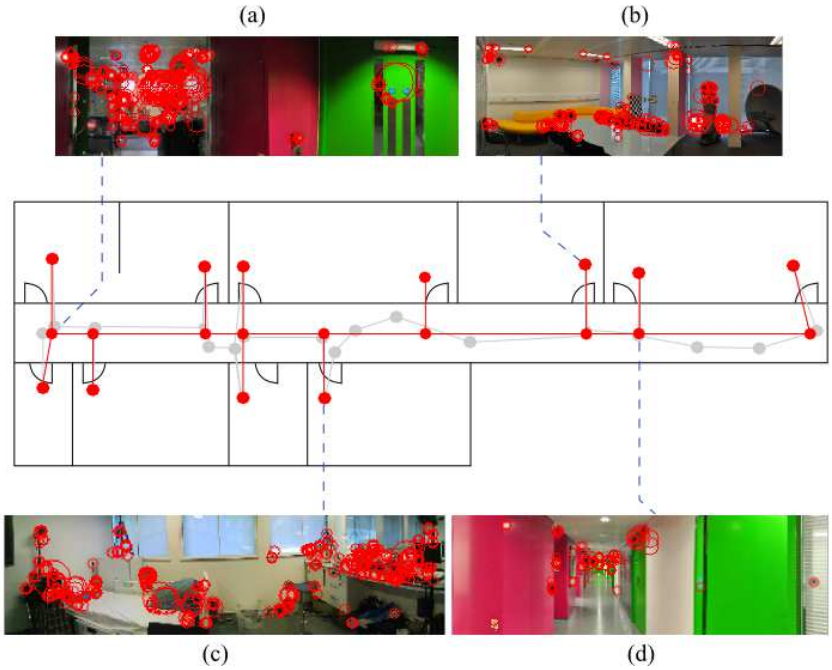


Fig. 12.14 Topological simplification for route planning. A simplified topological map of the environment, in comparison to Fig. 13.13, brings performance benefits for route planning and ease of visualization.

the 2D laser scanner to provide a more detailed view of the environment. This could help the robot identify the exact location of objects in the environment. Furthermore, a more detailed tracking system could help the robot to maintain the motion of all moving objects in the environment for improved planning and obstacle avoidance.

References

1. Ascani, A., Frontoni, E., Mancini, A., Zingaretti, P.: Feature group matching for appearance-based localization. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pp. 3933–3938 (2008)
2. Bauer, A., Klasing, K., Lidoris, G., Mühlbauer, Q., Rohrmüller, F., Sosnowski, S., Xu, T., Kühnlenz, K., Wollherr, D., Buss, M.: The autonomous city explorer: Towards natural human-robot interaction in urban environments. International Journal of Social Robotics **1**(2), 127–140 (2009)
3. Bellotto, N., Hu, H.: Multisensor integration for human-robot interaction. The IEEE Journal of Intelligent Cybernetic Systems **1** (2005)
4. Bellotto, N., Hu, H.: Multisensor-Based Human Detection and Tracking for Mobile Service Robots. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics **39**(1), 167–181 (2009)
5. Bücken, S.: Learning Maps for Indoor Mobile Robot Navigation **99**, 21–71 (2008)

6. Buhmann, J., Burgard, W., Cremers, A., Fox, D., Hofmann, T., Schneider, F., Strikos, J., Thrun, S.: The mobile robot Rhino. AI Magazine **16**(2), 31 (1995)
7. Busch, M., Blackman, S.: Evaluation of IMM filtering for an air defense system application. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 2561, pp. 435–447 (1995)
8. Chakravarty, P., Jarvis, R.: Panoramic vision and laser range finder fusion for multiple person tracking. In: Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pp. 2949–2954 (2006)
9. Fritsch, J., Kleinehagenbrock, M., Lang, S., Plötz, T., Fink, G., Sagerer, G.: Multi-modal anchoring for human–robot interaction. Robotics and Autonomous Systems **43**(2-3), 133–147 (2003)
10. Hasanuzzaman, M., Ampornaramveth, V., Zhang, T., Bhuiyan, M., Shirai, Y., Ueno, H.: Real-time vision-based gesture recognition for human robot interaction. In: IEEE International Conference on Robotics and Biomimetics, ROBIO, pp. 413–418 (2004)
11. Ishii, Y., Hongo, H., Yamamoto, K., Niwa, Y.: Real-time face and head detection using four directional features. In: Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Proceedings, pp. 403–408 (2004)
12. Jin, Y., Mokhtarian, F.: Towards robust head tracking by particles. In: IEEE International Conference on Image Processing, ICIP, vol. 3, pp. 864–867 (2005)
13. Koenig, N.: Toward real-time human detection and tracking in diverse environments. In: IEEE 6th International Conference on Development and Learning, ICDL, pp. 94–98 (2007)
14. Krotosky, S., Cheng, S., Trivedi, M.: Real-time stereo-based head detection using size, shape and disparity constraints. In: IEEE Intelligent Vehicles Symposium, Proceedings, pp. 550–556 (2005)
15. Lee, H., Kim, J.: An HMM-based threshold model approach for gesture recognition. IEEE Transactions on pattern analysis and machine intelligence **21**(10), 961–973 (1999)
16. Li, F., Kosecka, J.: Probabilistic location recognition using reduced feature set. In: IEEE International Conference on Robotics and Automation, pp. 3405–3410. Citeseer (2006)
17. Li, Y., Ai, H., Huang, C., Lao, S.: Robust head tracking based on a multi-state particle filter. In: Automatic Face and Gesture Recognition, 7th International Conference on, pp. 335–340 (2006)
18. Lindstrom, M., Eklundh, J.: Detecting and tracking moving objects from a mobile platform usinga laser range scanner. In: 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, Proceedings, vol. 3, pp. 1364–1369 (2001)
19. Loper, M., Koenig, N., Chernova, S., Jones, C., Jenkins, O.: Mobile human-robot teaming with environmental tolerance. In: Proceedings of the 4th ACM/IEEE international conference on Human robot interaction, pp. 157–164. ACM (2009)
20. Lowe, D.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
21. Luo, J., Savakis, A., Singhal, A.: A Bayesian network-based framework for semantic image understanding. Pattern Recognition **38**(6), 919–934 (2005)
22. Meyer, J., Filliat, D.: Map-based navigation in mobile robots: II. A review of map-learning and path-planning strategies. Cognitive Systems Research **4**(4), 283–317 (2003)
23. Montemerlo, M., Roy, N., Thrun, S.: Perspectives on standardization in mobile robot programming: The Carnegie Mellon navigation (CARMEN) toolkit. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 2436–2441. Citeseer (2003)
24. Nickel, K., Stiefelhagen, R.: Real-Time Person Tracking and Pointing Gesture Recognition for Human-Robot Interaction. In: Computer vision in human-computer interaction: ECCV 2004 Workshop on HCI, Prague, Czech Republic, proceedings, p. 28. Springer-Verlag New York Inc (2004)
25. Reynolds, C.: Steering behaviors for autonomous characters. In: Game Developers Conference. http://www. red3d. com/cwr/steer/gdc99. Citeseer (1999)
26. Schaffalitzky, F., Zisserman, A.: Automated scene matching in movies. Lecture notes in computer science pp. 186–197 (2002)

27. Schmid, C., Mohr, R.: Local grayvalue invariants for image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(5), 530–535 (1997)
28. Shalom, Y., Blair, W.: Multitarget-Multisensor Tracking: Applications and Advances. Boston: Artech House **3** (2000)
29. Shan, C., Wei, Y., Tan, T., Ojardias, F.: Real time hand tracking by combining particle filtering and mean shift. In: Sixth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 669–674. Citeseer (2004)
30. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (1994)
31. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. ICCV, vol. 2, pp. 1470–1477. Citeseer (2003)
32. Thrun, S.: Exploration and model building in mobile robot domains. In: Proceedings of the IEEE International Conference on Neural Networks, vol. 1, pp. 175–180. Citeseer (1993)
33. Topp, E., Christensen, H.: Tracking for following and passing persons. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 70–76. Citeseer (2005)
34. Ullah, M., Pronobis, A., Caputo, B., Luo, J., Jensfelt, P., Christensen, H.: Towards robust place recognition for robot localization. In: Proc. IEEE IntŠl Conf. Robotics and Automation, pp. 530–537. Citeseer (2008)
35. Valibeik, S., Yang, G.: Segmentation and Tracking for Vision Based Human Robot Interaction. In: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 03, pp. 471–476. IEEE Computer Society Washington, DC, USA (2008)
36. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision **57**(2), 137–154 (2004)
37. Wimpey, B., Drucker, E., Martin, M., Potter, W.: A Multilayered Approach to Location Recognition. Proc. SoutheastCon pp. 1–7
38. Won, W., Kim, M., Son, J.: DriverŠs Head Detection Model in Color Image for DriverŠs Status Monitoring. In: Intelligent Transportation Systems, 11th International IEEE Conference on, pp. 1161–1166 (2008)
39. Zhang, Q., Pless, R.: Extrinsic calibration of a camera and laser range finder (improves camera calibration). In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Proceedings, vol. 3, pp. 2301–2306 (2004)
40. Zhang, Z.: Flexible camera calibration by viewing a plane from unknown orientations. In: International Conference on Computer Vision, vol. 1, pp. 666–673 (1999)
41. Zhang, Z., Gunes, H., Piccardi, M.: An accurate algorithm for head detection based on XYZ and HSV hair and skin color models. In: 15th IEEE International Conference on Image Processing, ICIP, pp. 1644–1647 (2008)
42. Zhu, Y., Ren, H., Xu, G., Lin, X.: Toward real-time human-computer interaction with continuous dynamic hand gestures. In: Proc. Fourth Int. Conf. Autom. Face Gesture Recogn., Grenoble, France, IEEE Comput. Soc, pp. 544–549 (2000)