University of Alberta

Gabor-gist Visual Homing

by

Michael Mills

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Masters of Science

Department of Computing Science

Fall 2013
Edmonton, Alberta

To Mom
Beause she couldn't be here to see it.

Abstract

Many robotic systems are required to navigate or home to learned location using minimal resources. Autonomous robots are generally limited in computation and storage resources, imposing significant challenges on algorithm design. Particularly when only visual data is used, these algorithms need to be robust and efficient. In addition, independence from a scene model is preferred. Extraction of models and calibration procedures are time consuming and sensitive to changes in the environment. Visual homing without a geometric model is studied in mapless or qualitative visual homing. In this thesis, we adopt a framework based on View-Sequenced Route Representation (VSRR) and contribute in two areas: Compact representation of the path and visual homing along a desired route using the representation, and secondly develop an algorithm which localizes the robot using a novel concept we call *eigensegments*. The effectiveness of the system is demonstrated with both indoor and outdoor environments.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Autonomous and semi-autonomous robots have a broad range of applications from planetary space exploration to household cleaning. A common attribute of such applications is that the robot needs to travel between two previously visited locations. Visual homing deals with navigating between locations but, faces significant challenges due to limitations in on board resources, sensing modalities, and robot actuation.

Mobile robots typically do not have the resources required to perform 3D reconstructions of an environment. It is possible to construct 3D scene model and use it for homing; however, it typically requires large computational and storage resources. It is, therefore, desirable to have mapless qualitative algorithms and control architectures that are free of 3D models. Sensory input is often corrupted by measurement noise, outliers and uncertainties in actuation, requiring algorithms robust, efficient and practical for resource limited setups. Qualitative methods relax strict constraints of 3D models and makes decisions based on a qualitative measure.

The basic problem this thesis addresses is mapless qualitative visual homing using a compact path representation. Specifically, visual homing in environments without obstacles or visual occlusions. A novel visual homing method is developed using the Gabor-gist descriptor of [45] and the View-Sequence Route Representation framework of [29].

In Chapter 2 a background of visual homing in robotics is presented. The general overview of the literature is divided into two main classes of algorithm, namely keypoint and whole image methods. Chapter 3 presents the proposed method, divided into a training phase in Section 3.1 and replay phase in Section 3.2. The training phase details how to construct a visual path using the Gabor-gist descriptor. While the replay phase covers how this visual path representation is used to autonomously retrace the path. Chapter 4 presents experiments and evaluations, divide experiments into indoor environments in Section 4.3.1 and outdoors in Section 4.3.2.

The contributions of this thesis are: (1) A new compact representation of a path using the Gabor-gist descriptor. (2) A novel method for representing a segment of the visual path using Gabor-gist and eigenvector's or principal components.

Chapter 2

Background and Literature Review

2.1 Overview of Visual Homing

This chapter reviews the visual homing literature and formulates the visual homing problem. Appearance-based homing methods are divided into three main categories: Keypoint, Machine learning and Whole Image methods. Each category is discussed using the available homing algorithms to motivate why mapless methods are an important area of research and the storage problems this thesis addresses.

Section 2.1.3 refers to methods using SIFT, SURF, etc. but only in a qualitative sense, and not performing a metric reconstruction of the environment. Section 2.1.4 presents methods which use machine learning techniques to form a mapping between images and heading corrections. These methods do not generally use a sequence of keyframes to perform the homing task. Finally Section 2.1.5 refers to methods using whole images as keyframes, comparing these with the current view to derive a heading correction and localize the robot.

2.1.1 Visual Odometry

Many of today's robots, such as the Mars rovers [39, 6], perform visual navigation tasks such as homing using visual odometry. In navigation, traditional odometry is the use of data from the movement of actuators to estimate change in position over time.

While useful, traditional odometry techniques suffer from precision problems because wheels tend to slip and slide on any surface creating a non-uniform distance traveled compared to the wheel rotations. Visual odometry is the process of determining equivalent odometry information using sequential images to estimate the distance traveled and, allows for enhanced navigational accuracy in robots or vehicles on any surface.

A proven technique, visual odometry typically makes heavy use of local image features to extract the metric measurement of change in position [39]. An alternative is the "direct" or appearance-based visual homing techniques, which seek to minimize errors directly in sensor space. These methods generally avoid feature extraction, matching, tracking and the costly 3D reconstruction and estimations.

Further these methods can be made more compact requiring less storage space than other methods.

2.1.2 Appearance-based Homing

Appearance-based strategies consist of two procedures. First, a training phase where images or prominent features of the environment are recorded and stored as templates. The templates are labeled with certain localization information and/or with an associated control steering commands. Second, an autonomous navigation stage, where the robot has to recognize the environment and self-localize in it by matching the current view with the stored templates. The main problems of appearance-based strategies are finding an appropriate algorithm to create the environment representation and defining the view matching criteria.

Deviations between the route in the training phase and the route navigated in the replay phase yield different sets of images, and thus differences in the perception of the environment. Contributions have focused mainly on improving the way images are recorded and then matched in the replay phase. There are two main approaches for environment perception and recognition without using a map:

1. Model-based Approaches. These approaches use pre-defined object models to recognize features in complicated environments and self-localize.

2. View-based Approaches. The self-localization is performed using image matching algorithms.

In this thesis we concentrate on the view-based approach. Matsumoto's work [29] presents a model which is capable of both localization and steering angle determination simultaneously using standard pixel images without the need for predefined models. In his work [29] Matsumoto proposed a visual representation of the route, the View-Sequenced Route Representation (VSRR). The VSRR is a non-metrical model of the route, which contains a sequence of front view images along a route memorized in a training run creating a visual path. In the autonomous run the two types of both localization and steering angle are achieved in real-time by matching between current view I_t and the memorized view sequence using a correlation technique.

A visual path is defined by arranging keyframes I_i in a sequence, termed the view sequence. We acquire the view sequence in a training run, thus dividing the path into segments, each with a keyframe I_i. The entire path can be seen as a sequence of segments:

$$path \ = \ \{I_0, ...I_{N-1}\} \tag{2.1}$$

where I denotes the keyframe is an intensity image and i the index within the view sequence, and N is the number of segments in the path. Building a view sequence is done through a simple algorithm.

1. i=0

2. Save the current view I_t as keyframe I_i where i is the current keyframe.

3. Move the robot forward until current view I_t changes to a certain degree with respect to the latest keyframe I_i.

4. $i = i + 1$.

5. goto 1.

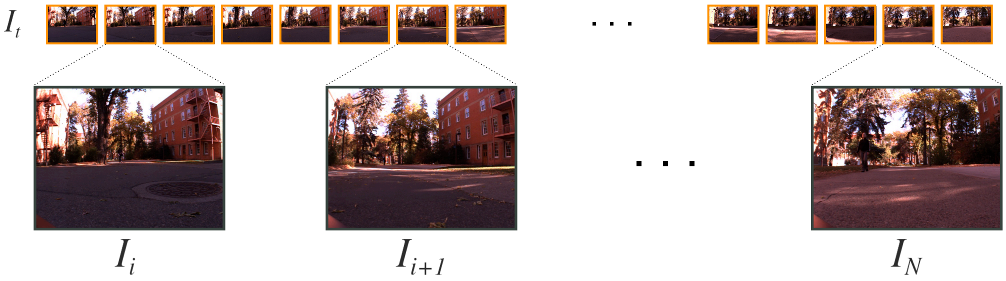


Figure 2.1: The construction of the visual path from a stream of images. Keyframes are selected by comparing the current image I_t with the latest keyframe I_i.

During the autonomous replay phase I_t is compared to the current keyframe I_i and the calculation of similarity is done. In [29] this process was achieved by template matching, using the central rectangle portion of one image as the template. As a result of this matching process, the horizontal displacement value of the template is acquired and used as a steering correction signal. Matsumoto's work demonstrates the ability to retrace a route using images and odometry information. Specifically it is shown that a view sequence consisting of images has the necessary information for localization, and steering angle determination.

Matsumoto presented the view-sequence as a sequence of images. A more general framework is to view the path as a sequence of *segment's*. A segment represents a portion of the path rather than a single goal location. Segments can be represented as single views or keyframes, but also by a set of keypoint's, image's, image descriptor's or any description that characterizes the images captured within the segment.

2.1.3 Keypoint Methods

Keypoints while prone to matching errors are proven to be effective at recognizing objects. Treating a scene as an object to be recognized some authors propose a simple approach for visual homing. These algorithms like [29] are qualitative in nature, requiring no explicit map of the environment, nor image Jacobian, homography, or fundamental matrix. Keypoint image coordinates are compared with those obtained during the training phase in order to determine heading corrections. These systems require single off-the-shelf forward-looking cameras with no calibrations. Keypoints also provide some invariance to rotation and partial scene occlusion giving these methods the ability to operate both indoors and outdoors as well as on flat, slanted, and rough terrain with dynamic occluding objects.

As opposed to whole images keypoint methods [5, 3, 40] make use of features such as SIFT[26], SURF[4], etc. Keypoints are defined in terms of a neighborhood around a point of interest, a procedure commonly referred to as feature extraction. Keypoint detection produces local decisions as to whether there is a feature at a given image point. A descriptor is a description of an image pattern around a keypoint. It is usually associated with a change of an image property or several properties simultaneously. Image properties commonly considered are intensity, color, and texture. The descriptors can then be used for various applications, but here we are concerned with matching and aligning images.

Features have proven to be a powerful tool in mapless homing [5, 3, 40]. Keypoints must be extracted in the image stream and matched with a keyframe to design the control law. Robust extraction and real-time tracking or matching of these visual cues is a nontrivial task and a bottleneck of a real-time system [29, 28, 21, 38, 5].

2.1.4 Machine Learning Methods

An alternate mapless approach is to learn the mapping from images to turning commands based on their classification [1]. Ackerman introduces a method for rapidly classifying visual scenes globally along a small number of navigationally relevant dimensions: depth of scene, presence of obstacles, path versus non-path, and orientation of path. They show that the algorithm reliably classifies scenes in terms of these high-level features, based on globally localized spectral analysis similar to early-stage biological vision. They demonstrate successful training and subsequent autonomous path following for two different outdoor environments. However, these methods rely on an unstable learning algorithms.

Another approach that has received considerable attention [11, 48, 50, 42, 23, 46, 18] is to store an example image with each specific location of interest. At run time, the image database is searched to find the image that most closely resembles the current one (or, alternatively, the current image is projected onto a manifold learned from the database [24, 33]). Such approaches require extensive training and have difficulty providing sufficient spatial resolution to determine actual turning commands in large environments. Similarly, sensory-motor learning has been used used to map visual inputs to turning commands, but the resulting algorithms have been too computationally demanding for real-time performance [13].

Mateus Mendes[31] proposes the visual path be encoded in a data structure other then a sequence of keyframes. Many authors agree that the source of intelligence is, to a large extent, the use of a huge memory [17, 22, 2]. It has been proposed that sequences of events which guide our later behaviour are stored in an associatative memory structure. Inspired by that idea, Mendes [31] controls a robot using sequences of images stored in a Sparse Distributed Memory a kind of associative memory based on the properties of high dimensional binary spaces - which theoretically exhibits some human-like properties. J. Hawkins [17] proposes the Memory Prediction Framework, modeling the brain as continuously making predictions about the environment. When a prediction is violated, adjustments in the brain's memories are made according to the new data. This memory appears to be organized in a hierarchy, each level responsible for learning only a portion of the overall model. Kanerva [22] proposed a model to implement this prediction framework, known as a Sparse Distributed Memory (SDM). It is designed for storing and retrieving large amounts of information without focusing on the accuracy of the information. It uses input patterns as memory addresses, where information is retrieved based on similarities between these addresses and thus patterns.

Localization is calculated based on the similarity of two views: A keyframe I_i and the current view I_t. Whichever view is returned by the SDM is essentially the keyframe used for heading correction. To calculate the heading correction error a window search process is used. A search window selected from the center of the returned keyframe image is matched against an equivalently sized window in the current view, calculating the horizontal displacement that results. The robot shows good ability to correctly follow most of the sequences learned, with small errors and immunity to the kidnapped robot problem.

A limitation of this approach is that of sensitivity to image noise and illumination changes. A furthur drawback is the structure only requires storage of about 0.1 bits per bit, limiting the scalability of the approach. Currently this method shows promise but requires a large overhead of both computation and storage, computation in regards to processing of images into the memory structure and storage of pixel images in that memory, limiting the scalability of this method.

2.1.5 Whole Image Methods

Keypoint's have benefits, some authors however propose new ways of comparing images. Matsumoto's original work [29] uses a cross correlation to measure similarity, others have proposed new more robust methods. Dame [8] proposes to use entropy, they show that it is possible to navigate along a visual path without relying on the extraction, matching or tracking of keypoints. The proposed approach relies directly on the information contained in the image. Dame shows that it is possible to build a control law directly from the maximization of the mutual information between a current image and the current keyframe. Mutual information has been shown to be robust to illumination variations and occlusions [7, 36]. As a result, the need for the generally complex task of keypoint extraction and matching is eliminated.

The primary drawback of this approach and other similar methods is the requirement of saving pixel images, and the storage requirements this entails. The storage space requirements are multiplied by the need for a dense sampling of the visual path. High frequency of keyframe sampling roughly three images/m is required for Dame's method [8]. For a route of 100 m 300 images are required to encode the visual path. If images are on the order of even 320x240 for a 100 m route 14 MB of images are required, even if the images are compressed.

In this thesis we will demonstrate it is possible to define a control law directly linked to a similarity maximization while eliminating the need of for saving pixel images. We show that the visual path can be encoded using a whole image descriptor known as Gabor-gist, reducing the storage requirements to a few hundred bytes/keyframe.

2.2 Gabor-gist image descriptor

Gist refers to the meaningful information that an observer can identify from just a glimpse of a scene [35]. The idea of gist is to define what a "scene" is, as opposed to an "object" or "texture" within a scene. Gist essentially attempts to capture the spatial arrangement of individual elements. Keypoint feature's (SIFT, SURF, etc) are primarily designed to recognize objects within a scene. While keypoint's have been demonstrated as useful for tasks such as homing, we demonstrate that visual homing can be accomplished by capturing *gist* of a scene. There exist numerous methods of extracting the *gist* of a scene [41, 14, 19, 12, 34, 32, 16, 43, 37, 45, 10, 35, 40]. Here we concentrate on the Gabor-gist method of [44, 45].

When viewing a scene for a short time, humans extract enough visual information to accurately recognize its categorical properties (e.g., trees on a mountain side). Most of the information concerning individual objects and their locations are overlooked, rather viewing a scene as having it's own shape, carrying its identity. Object categories like cars or animals, look alike because they have the same "function", Oliva [35] showed that scenes belonging to the same category share a similar and stable spatial structure (shape) that can be extracted and used to classify a scene into categories (e.g. Outdoors, Indoors, etc). They show that perceptual properties exist that can be uncovered using simple computations, and that these properties can be translated into a meaningful description of the scene shape.

The *gist* description of a scene is useful beyond scene classification. Torralba and Oliva expand their *gist* methodology to the estimation of depth from image structure. They demonstrate that, by recognizing the properties of the structures present in the image, they can infer the scale of the scene and, therefore, its absolute mean depth [44]. They expand their work to place and object recognition [45]. In both [44, 45] they use a wavelet image decomposition, where each image location is represented by the output of filters tuned to different orientations and scales.

The representation of an image is given by a collection micro-feature statistics. Here the collection of micro-features are the responses to a set of Gabor filters $h_k(\boldsymbol{x})$ convolved with a pixel image I [1].

[1] Here we only consider gray scale images but the same technique can be applied to RGB channels.

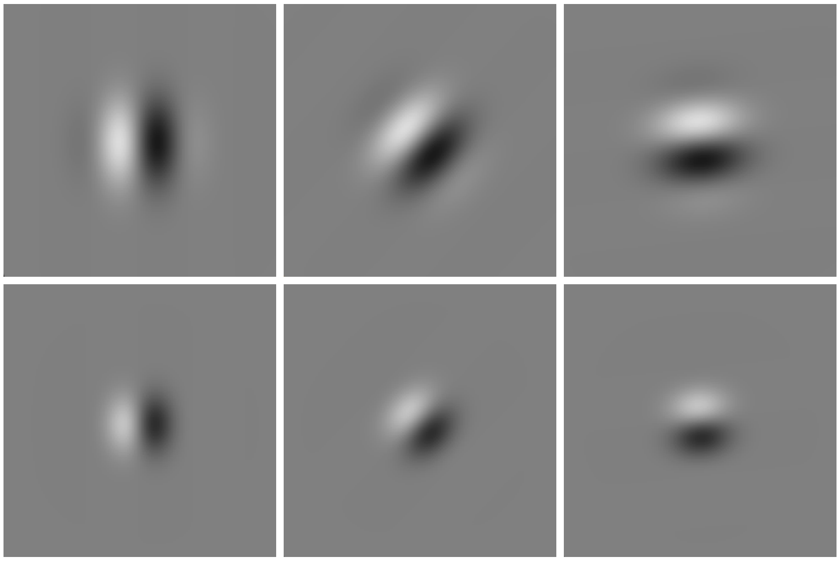


Figure 2.2: Gabor filters tuned to different scales and orientations. Each row represents a scale and, each column an orientation.

$$h_k(\boldsymbol{x}) \quad = \quad \left(\frac{1}{2\pi\sigma_x\sigma_y}\right) exp\left[-\frac{1}{2}\left(\frac{\widetilde{x}^2}{\sigma_x^2}+\frac{\widetilde{y}^2}{\sigma_y^2}\right)\right] exp\left[2\pi j W\widetilde{x}\right] \tag{2.2a}$$

$$\begin{aligned} \widetilde{x} &= \ x\cos\theta + \ y\sin\theta \\ \widetilde{y} &= \ -x\sin\theta + \ y\cos\theta \end{aligned} \tag{2.2b}$$

$$I_k(\boldsymbol{x}) = \sum_{\boldsymbol{x}'} I(\boldsymbol{x}')h_k(\boldsymbol{x}-\boldsymbol{x}') \tag{2.3}$$

$$v_t^L = \{I_k(\boldsymbol{x})\}_{k=1,N} \quad where\ N = \ \#\ of\ filters \tag{2.4}$$

$$m_t(\boldsymbol{x}, i) = \sum_{\boldsymbol{x}'} \left|v_t^L(\boldsymbol{x}')\right| w_i(\boldsymbol{x}-\boldsymbol{x}') \tag{2.5}$$

where $w_i(x)$ is an averaging window.

$$v_t^G = m_t(\boldsymbol{x}, i), i = 1, M \tag{2.6}$$

where M is the number of averaging windows, capturing the mean response value of a Gabor-filter within the window. These mean values capture the spatial relationships between image regions. The final result v_t^G is a concatenation of a grid of averages. (See figure 2.3).

Gabor filters h_k (eq 2.2) are interesting due to their connection to biological vision. Jones et al. have shown that Gabor filters are an accurate approximation

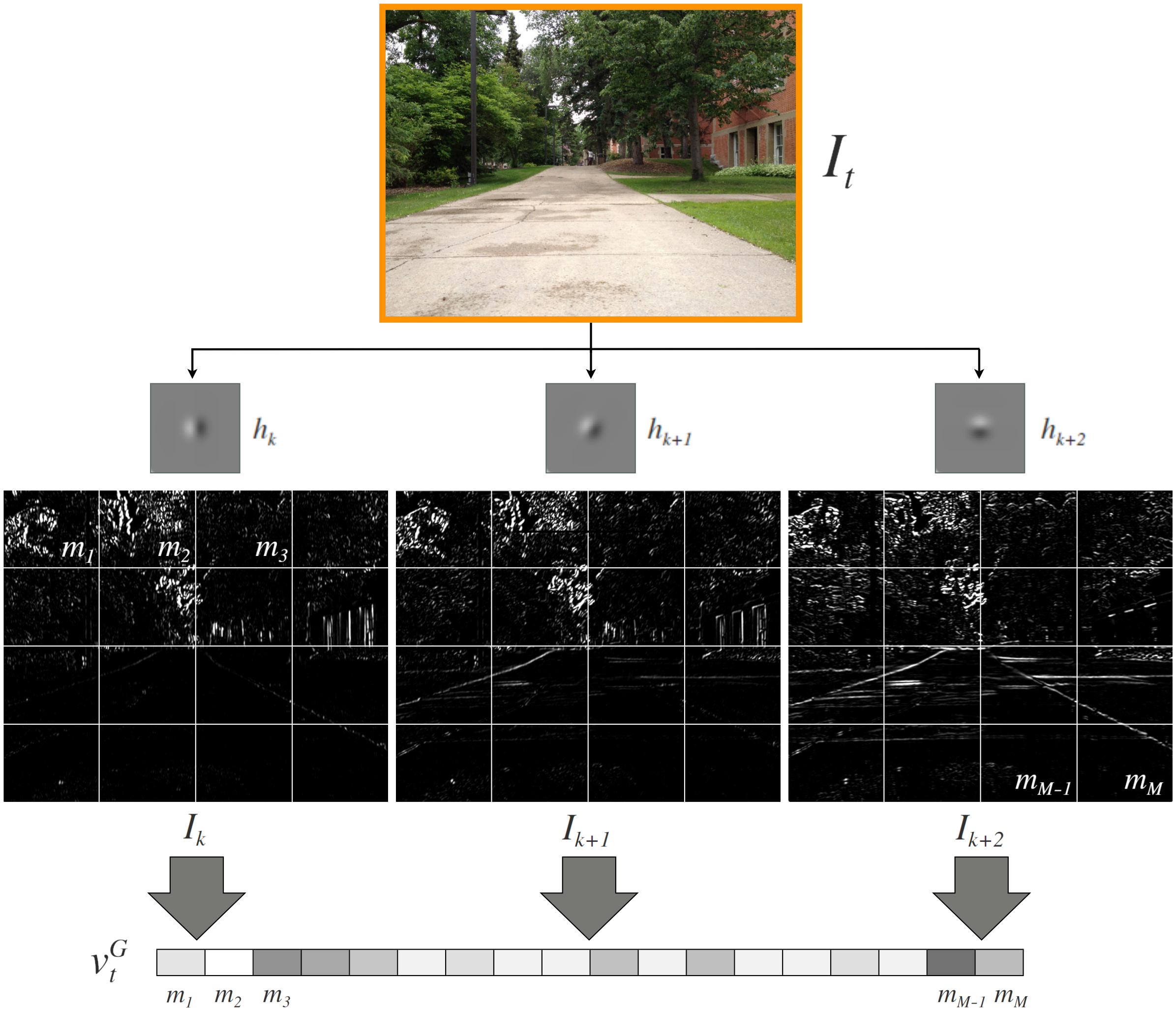


Figure 2.3: Overview of gist descriptor creation. An input image I_t is convolved with a bank of Gabor filters h_k. A mean value is extracted from the cells of a grid placed over each response image I_k. The mean values are concatenated into the final vector v_t^G.

of neural response patterns in the mammalian visual cortex [20]. Our understanding of visual information processing in the mammalian cortex has been dominated by neurons which respond to narrow ranges of stimulus orientation and spatial frequency [20]. The 2D Gabor filters flexibility (ie. continuous nature) may confer advantages on the system that employs them because the parameters are continuous, a system can be fine-tuned according to the environment, either through early visual experience [20] or by continual reconfiguration [27]. Gabor representations have been shown to be optimal in the sense of minimizing the joint two-dimensional uncertainty in orientation and frequency [9]. By examining the distributions of 2-D Gabor coefficients found by a neural network in different image regions, it is possible to achieve image segmentation on the basis of spectral signature.

Essentially these filters can be considered orientation and scale tunable edge detectors, the statistics of which, in a given region, are often used to characterize the underlying texture information [30]. The Gabor-gist image representation has proven to be effective at both depth estimation [44] and object detection and localization [45]. It has been applied to robotic localization and loop closure problems [40, 25]. It has yet to be applied to the problem of visual homing, where it is used to make navigation decisions towards a specific goal. This thesis will show that the Gabor-gist representation can be applied to the problem of visual homing.

Chapter 3

Gabor-gist Visual Homing

This chapter introduces the Gabor-gist visual homing method. The proposed method incorporates Gabor-gist into the general framework of VSRR (View Sequence Route Representation). The contributions of this chapter are twofold:

1. A visual path encoding using the Gabor-gist descriptor, segmenting the path into non-overlapping segments and encoding them using keyframes and eigenvectors.

2. A control system that utilizes the path encoding for both heading correction and localization along the path.

Section 3.1 details the training phase of the system, creation of a path representation. During this phase keyframes are selected from the image stream. Images not defined as keyframes are used to create a representation of the segment we term an *eigensegment*. In Section 3.2 we present the control system used to perform autonomous navigation. Heading correction in Section 3.2.1 responsible for adjusting the robots trajectory during homing. Second, present the localization of the robot along the path using *eigensegments*.

3.1 Training Phase

Using the VSRR framework established by [3, 15, 5, 8, 17] the robot is driven manually along the desired route processing images into a sequence of non-overlapping segments. Unlike previous approaches of storing pixel images or keypoints, the path is encode in segments as a compact Gabor-gist descriptor v_i^G and, a matrix of eigenvectors we have term an *eigensegment*.

The keyframe of a segment v_t^G is a Gabor-gist descriptor describing the goal location of the robot during replay. During training the current view v_t^G is compared with the current segment's keyframe v_i^G. When the similarity between the two descriptors (v_t^G and v_i^G) falls below an empirically determined threshold, a new segment is created and the process repeats. The result is a sequence of non-overlapping segments each with a keyframe v_t^G.

Previous methods do not generally consider non-keyframe images beyond the selection of keyframes. Our method uses these intermediate images to characterize a segment for localization. It occurred to us that images between keyframes are self similar but could vary in distinct ways from other segment's images. Further these variations could be used to recognize a segment. In the language of information theory we want to extract relevant information from a set of images in order to recognize them later. An approach is to capture the variation in a sequence of images, independent of any judgment of features. In mathematical terms, find the principal components of the distribution of images defining a segment, or the eigenvectors of the covariance matrix of the images gist descriptors. These eigenvectors can be thought of as features which together characterize a segment. Because these vectors are eigenvectors, and describe a path segment we call them an *eigensegment*.

An efficient way of learning and characterizing a segment by variation is Principal Component Analysis (PCA). In similar applications, PCA has been used to compare descriptors in a lower dimensional subspace saving computation time. However, performing comparisons in a reduced dimensional space compares only the information left after the projection to that subspace, like comparing two 3D objects by their 2D shadows. While the hope is to compare only those components that contain the most information, we have found this leads to decreased performance with respect to Gabor-gist (see Section 4.2).

Given v_i^G is gist descriptor of dimension N (classically N=320), these descriptors can be thought of as a point in an N dimensional space. Many such descriptors then map to a collection of points in an N dimensional space. Descriptors captured within a segment, being similar will not be distributed randomly within the descriptor space. The main idea of principal component analysis (PCA) is to find vectors, which best account for the distribution of these segment descriptors within the entire descriptor space. These vectors or principal components define the subspace of segment descriptors that best describe a segment.

Let $l = v_0^G, ..., v_M^G$ be a sequence of gist descriptors where M is the number of images captured during this segment's training. This set of descriptor vectors is then subjected to PCA which seeks a set of N orthogonal vectors and eigenvalues which best describe the distribution of descriptors l. The eigenvectors are then sorted by their eigenvalues, the top k of which form our *eigensegment* matrix U_i.

The resulting U_i matrix can then be viewed as a representation of the segment in terms of the components that best describe a segment's descriptor distribution. During the autonomous phase, this representation is used to localize the robot along the path. Figure 3.1 shows the entire process of creating a visual path.

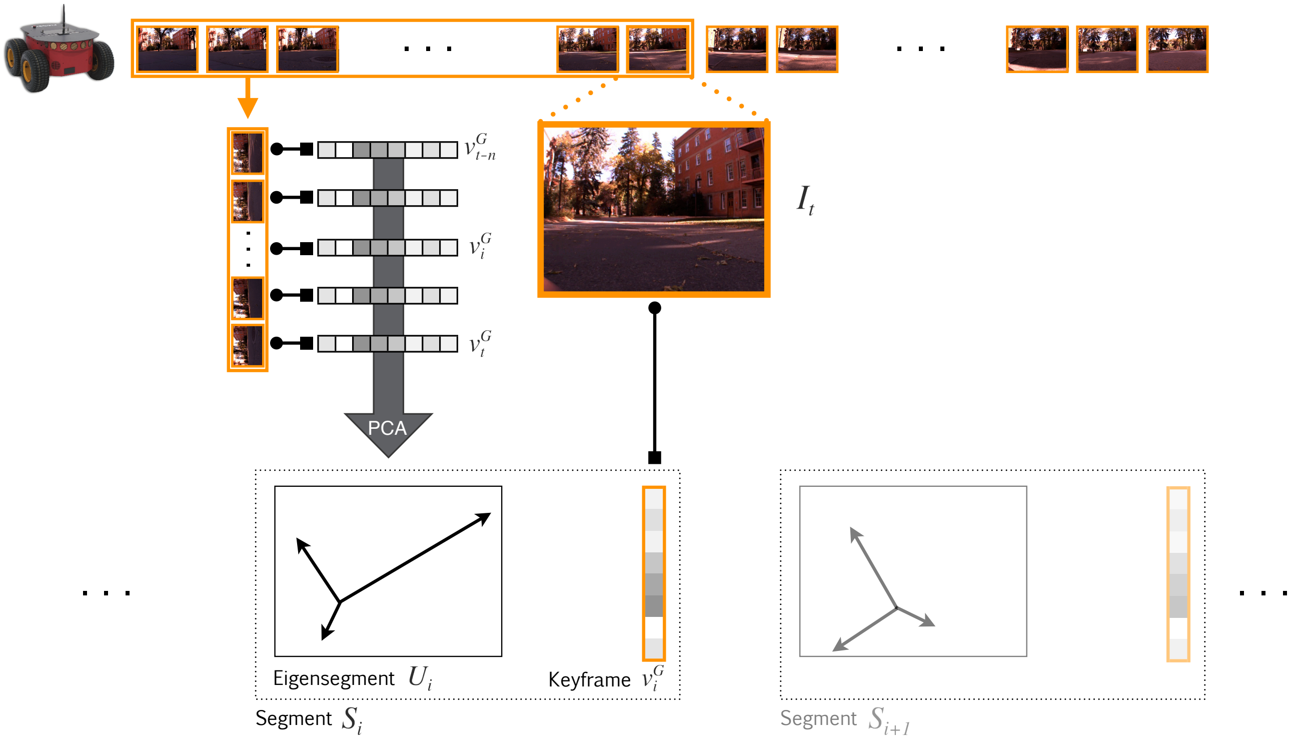


Figure 3.1: The training system builds a sequence of segments using the current view I_t. Each segment contains a gist description of the desired view or keyframe v_i^G and, a set of eigenvectors derived by PCA from images captured between keyframes U_i.

Algorithm 1 Gabor-gist training algorithm

```
$I \leftarrow capture(cam)$
$v_t^G \leftarrow gist(I)$
$keyframes \leftarrow \{v_t^G\}$
$frames \leftarrow \{\}$
$eigensegments \leftarrow \{\}$
$i \leftarrow 1$
$threshold \leftarrow 0.1$
$done \leftarrow False$
while $done == False$ do
    $I \leftarrow capture(cam)$
    $v_t^G \leftarrow gist(I)$
    if $(v_t^G \cdot keyframes[i]) > threshold$ then
        $keyframes \leftarrow keyframes + \{v_t^G\}$
        $(mean, eigenvectors) \leftarrow PCACompute(frames)$
        $eigensegments \leftarrow eigensegments + \{(mean, eigenvectors)\}$
        $frames \leftarrow \{\}$
    else
        $frames \leftarrow frames + \{I\}$
    end if
    $i \leftarrow i + 1$
    if $UserStop$ then $done = True$
    end if
end while
```

3.2 Replay Phase

In the replay phase, the robot proceeds autonomously and sequentially through the segments starting from approximately the same initial location as that of the training phase. The replay phase is broken into two components, heading correction and segment selection (ie. localization along the path). These two components run in parallel and at each time step we evaluate both heading corrections and segment selection. Heading correction adjusts the robot's heading, guiding it to the goal location. Segment selection localizes along the path beginning with the first segment.

3.2.1 Heading Correction

Heading correction using vision is essentially an image alignment problem, the robots current view v_t^G is aligned with the current segment's keyframe v_i^G. A similar window search to that of [8, 29, 31] has been modified for use with Gabor-gist. Performing a comparison at each location x, a gist descriptor is extracted from within the search window and compared to the current segment's keyframe v_i^G. We write this windowed gist as $v_t^G(I, x)$, where I is the input image and x the coordinates of the windows center.

Image coordinates x are computed with respect to a coordinate system centered at the principal point (i.e. the intersection of the optical axis and the image plane), resulting in positive coordinates on the right and negative coordinates on the left.

The search returns a set of similarity values associated with the position x of the search window. The results of a search can be graphed, showing the similarity of a window with respect to the horizontal position, as shown in Figure 3.2. The x position associated with the maximum similarity is the amount of shift required to align the two descriptors. x is then passed to the steering controller to execute the proper turning action, aligning towards the goal location [1].

$$x = \underset{x_i}{arg\,max} \left(1 - v_t^G(I, x) \cdot v_i^G\right) \tag{3.1}$$

passing x to a proportional controller, corrects the robots heading.

$$P_{out} = K_p x \tag{3.2}$$

where K_p is an empirically determined gain unique to the robot.

[1] The use of integral images make this search very efficient, requiring only around 100 floating point operations per search window.

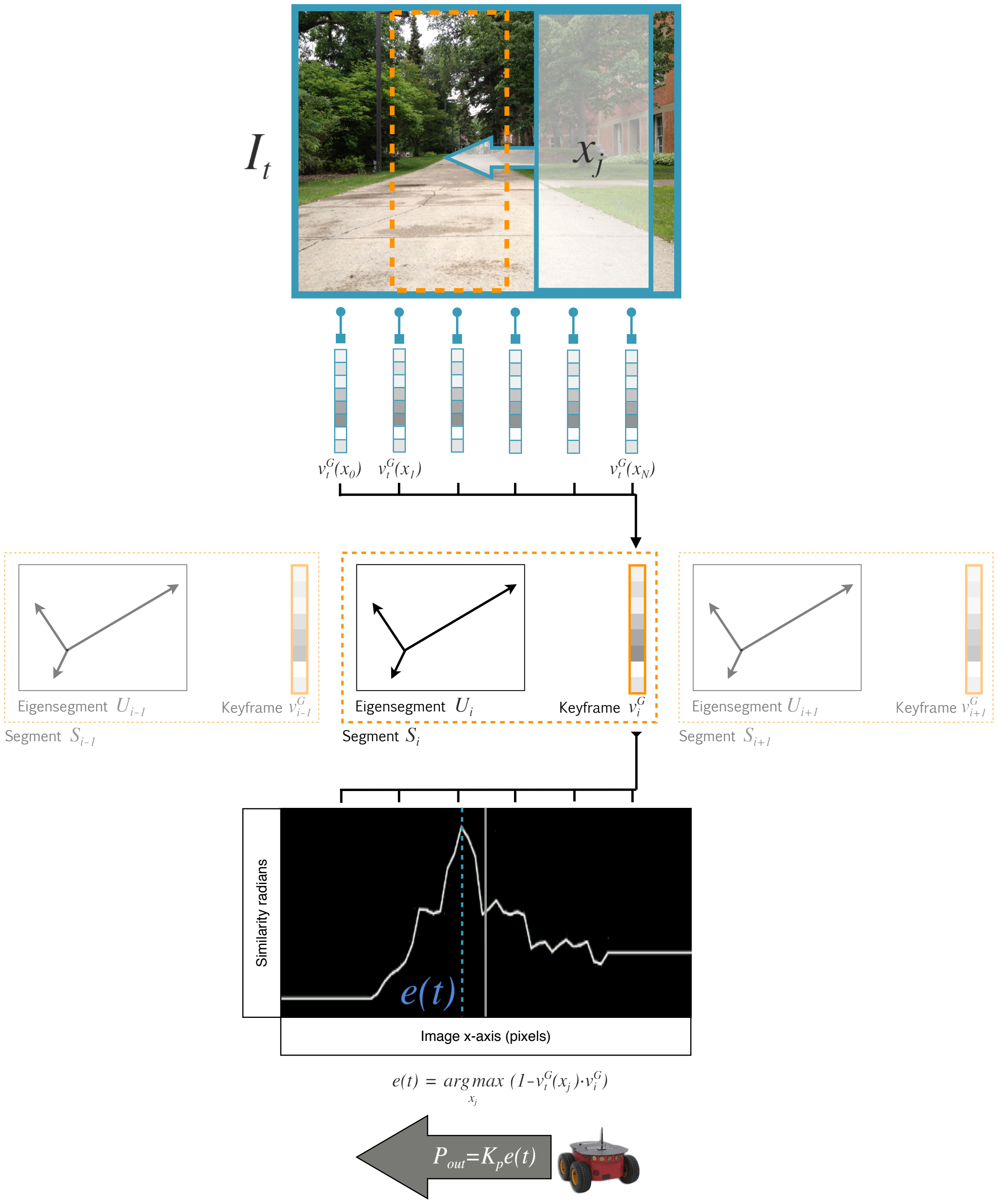


Figure 3.2: Heading correction. A search window scans the current view, comparing with the current segment's keyframe. The region with the greatest similarity defines the error between the current and desired heading. This error value is then passed to a proportional controller, controlling the wheels.

3.2.2 Path Segment Selection

Previous methods in selecting when to transition between segments are based on detecting when the robot has past a segment's goal location and should navigate towards the next goal. This has primarily been done by comparing the current view to the segment's keyframe, when a threshold in similarity is reached the robot is commanded to navigate based on the next segment in the sequence. Some methods incorporate odometry or other sensor data to improve performance. Such approaches have, however, proven difficult to extend and remain error prone. Segment transition errors are particularly hazardous to navigation because when an error occurs heading corrections are computed based on an incorrect assumption as to the robots location.

Each descriptor of a path image used to train an *eigensegment* can be represented by a linear combination of eigenvectors. The number of possible eigenvectors is equal to the descriptor size of gist. However a segment can also be approximated using only the "best" eigenvectors, or those that have the largest eigenvalues, which account for the largest variations. These eigenvectors are calculated by first subtracting the common elements then finding the largest varitions. To subtract the common elements the average descriptor of the training set is subtracted from the descriptors. This has the effect of leaving only the segment vartiations in visual structure. In segment recognition the common visual elements generally shared with many segments, the variations in a segment are what distinguish it from others. These large variations are generally not shared between segments, making them features capable of distinguishing segments.

The proposed method makes use of eigenvectors created using principal component analysis (PCA). An *eigensegment* characterizes a segment based on components which best represent the distribution of the segment's descriptors in the high dimensional space. PCA has traditionally been used to reduce the number of dimensions allowing for a faster comparison between descriptors. Our method uses PCA to measure an *eigensegment's* ability to reproduce the current view's descriptor. The current view descriptor is projected into a lower dimensional space and then reprojected back to the original space. This reprojection or reconstruction is then compared with the original descriptor, the *eigensegment* which best reconstructs the original descriptor is used as the current segment.

The following steps summarize the segment selection process:

1. Project the current view's descriptor v_t^G into several *eigensegments* lower dimensional spaces $v_{i \pm c}^U$, where c is a constant that defines a window of segments around the current segment i.

2. Reproject the descriptors $v_{i \pm c}^U$ back to the high dimensional space ${v_{i \pm c}^G}'$

3. Compare the original and reconstructed vectors and return the best reconstruction.

4. Set the current segment to the segment which best reconstructs the current view's descriptor.

Segments of a path consist of both a gist descriptor v_i^G and a set of eigenvectors in a matrix U_i. Both of these are stored in a sequence $\{(v_i^G, U_i), ..., (v_M^G, U_M)\}$. At each time step the robot's current view v_t^G is compared with the current segment

(v_i^G, U_i), as well as neighboring segments $(v_{i\pm c}^G, U_{i\pm c})$. The first step is to project the current views descriptor into an *eigensegment's* subspace.

$$v_t^U = U_i(v_t^G - \mu_i) \tag{3.3}$$

where μ_i the average gist descriptor of the the segment i. The result of this is a reduced gist vector v_t^U of dimension k, where $k <= N$ and N is the gist descriptor's original dimension.

We then attempt to reconstruct the original descriptor $v_t^{G'}$ by:

$$v_t^{G'} = (U^T \Phi_s) + \mu_i \tag{3.4}$$

The resulting vector $v_t^{G'}$ is of the same dimension as the original i.e. N, but is not an exact reproduction of the original descriptor because less than N eigenvectors are used. The reconstruction error e between v_t^G and $v_t^{G'}$ is determined using the cosine distance $e = v_t^G \cdot v_t^{G'}$. The *eigensegment* which minimizes the reconstruction error e becomes the current segment, and thus localizes the robot along the path. Figure 3.3 shows the entire process of selecting the next segment.

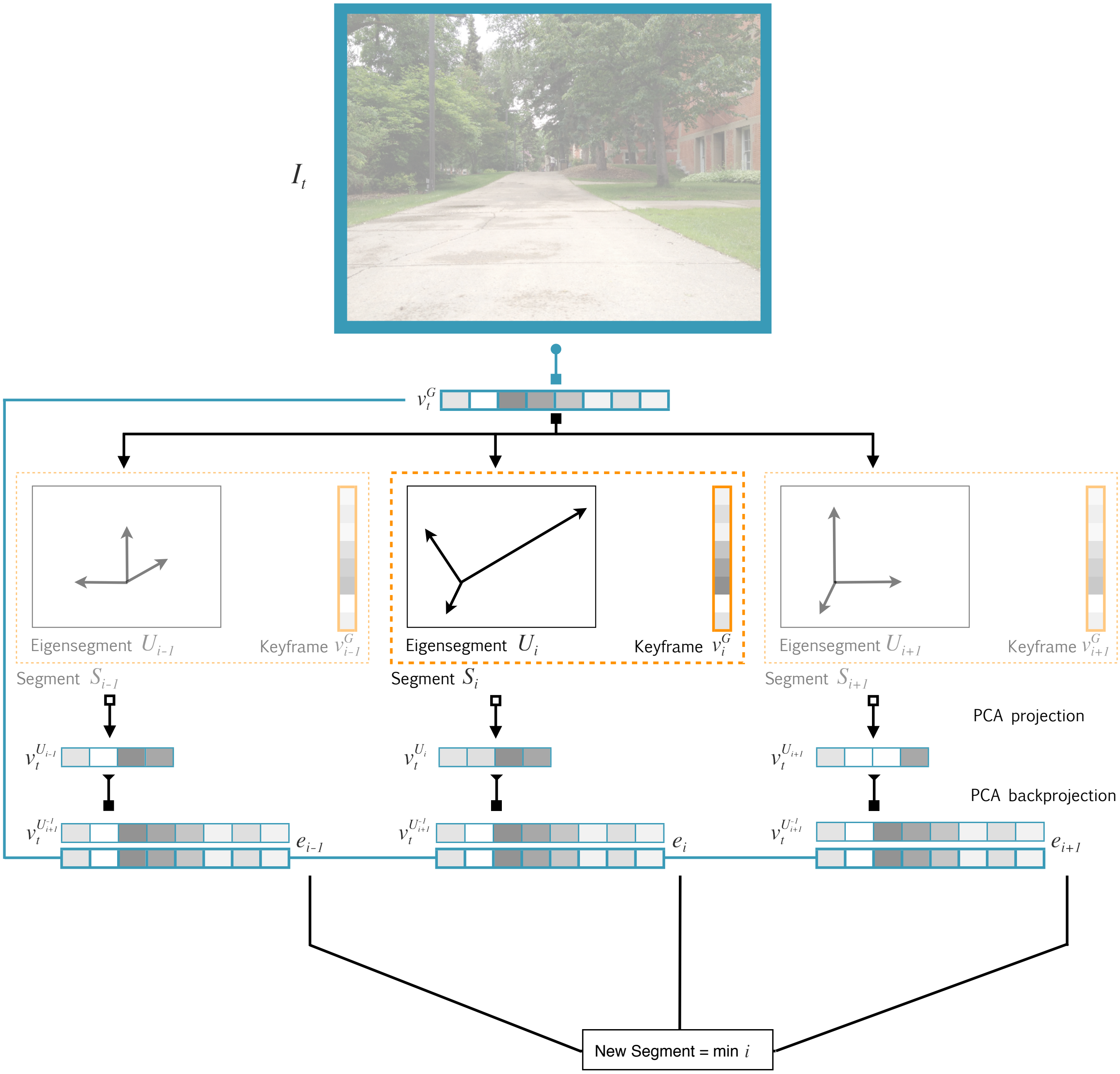


Figure 3.3: Selecting the next segment is done by projecting the current view into a range of *eigensegments* and then reprojected back. Which ever *eigensegment* does the accurately reconstructs the descriptor is selected as the current segment.

Algorithm 2 Gabor-gist replay algorithm

```
$keyframes \leftarrow \{load(keyframes)\}$
$eigensegments \leftarrow \{load(eigensegments)\}$
$finished \leftarrow False$
$i \leftarrow 1$ # current segment index
$\boldsymbol{x} \leftarrow \{x_0, x_1, ...x_n\}$ # Search window locations
while $finished == False$ do
    $I \leftarrow capture(cam)$
    $e(t) \leftarrow \underset{x_i}{arg\,max}\left(1 - v_t^G(I, \boldsymbol{x}) \cdot keyframes[i]\right)$
    $sims \leftarrow \{\}$
    for $j = (i-1)\,to\,(i+4)$ do
        $v_U^G \leftarrow PCAproject(v_t^G(I, e(t)), eigensegments[j])$
        $v_U^{G'} \leftarrow PCAreproject(v_U^G, eigensegments[j])$
        $sims[j-(i-1)] \leftarrow (1 - v_t^G(I, e(t)) \cdot v_U^{G'})$
    end for
    $i \leftarrow \underset{h}{arg\,min}\,(sims[h])$
    if $i >= |keyframes|$ then
        $finished \leftarrow True$
    end if
end while
```

3.3 Summary

The preceding section details the Gabor-gist visual homing algorithm. Section 3.1 covers training of the visual path. In training an operator manually drives the robot along the path. Images captured are processed using Gabor-gist into segments, each consisting of a keyframe and a set of eigenvectors. Keyframes are selected by comparing the latest segments keyframe to the current view, when the similarity drops below a threshold a new segment is created. In the creation of a new segment image descriptors captured since the previous keyframe are analyzed using PCA into an *eigensegment*.

Section 3.2 presents the autonomous replay of a learned visual path. In the replay phase the robot is placed at the paths starting location. As the robot replays the path each image captured are used to both correct the heading and localize the robot along the path. Heading correction performs a search of the current image for the region most similar to the current keyframe. The horizontal distance from the image center to the most similar region is passed to a proportional controller, controlling the drive. Localization using *eigensegment's* is accomplished through a reconstruction of the current views descriptor. The current segment is selected by comparing the ability of a set of *eigensegments* to reconstruct the current view's descriptor, the *eigensegment* with the best reconstruction is selected as the current segment, localizing the robot.

Experimental results and evaluations are presented in Chapter 4. The results show that the proposed method is promising and can be applied to a real-time robotic system.

Chapter 4

Experiments

4.1 Experimental Setup

To demonstrate Gabor-gist visual homing experiments were conducted both indoors and outdoors. Further simulations to validate the *eigensegment* method of segment selection were conducted. Only a single monocular camera has been used, no other sensor data such as GPS, radar, odometry are considered and the 3D structure of the scene remains fully unknown. No obstacle avoidance is considered therefore the navigation tasks have been performed in quiet conditions. Nevertheless several people do appear in the camera's view despite this, and thanks to the robustness of Gabor-gist descriptor, the navigation tasks completed successfully.

The experimental setup is built using ROS (Robotic Operating System), created in the Stanford artificial intelligence laboratory, and further developed by Willow Garage. ROS provides libraries and tools to help software developers create robotic applications. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, and package management. ROS is licensed under an open source, BSD license [47].

The experimental setup consists of these hardware components:

- Monocular Point Grey Dragonfly camera (640x480 30fps)
- Pioneer P3-AT Robot platform (Differential drive)
- MacBook Pro (OS X)

ROS configuration:

- Camera Node: Captures and sends images to other nodes
- Gist Homing Node: Calculates heading corrections from captured images
- Controller Node: Uses heading corrections to control the robot
- Pioneer Node: Controls the robot's drive system

4.2 Eigensegment Experiments

This section establishes the feasibility of the *eigensegment* as a replacement for keyframe comparisons. Simulations consisted of both indoor and outdoor datasets each containing a loop. Some sample images are shown in Figure 4.1. Each environment poses separate challenges. Indoor environments tend to produce a great deal of visual aliasing, where two different locations share similar visual characteristics. Outdoor environments present challenges in terms changes in lighting and dynamic components such as people moving through the scene.

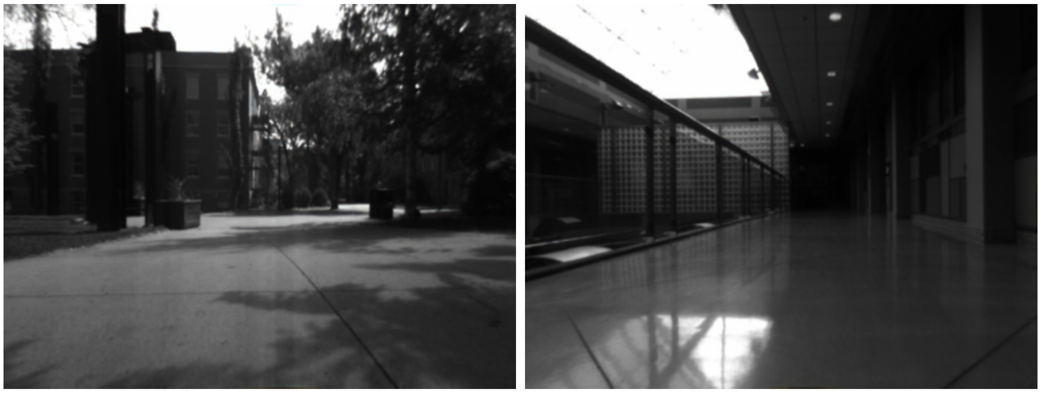
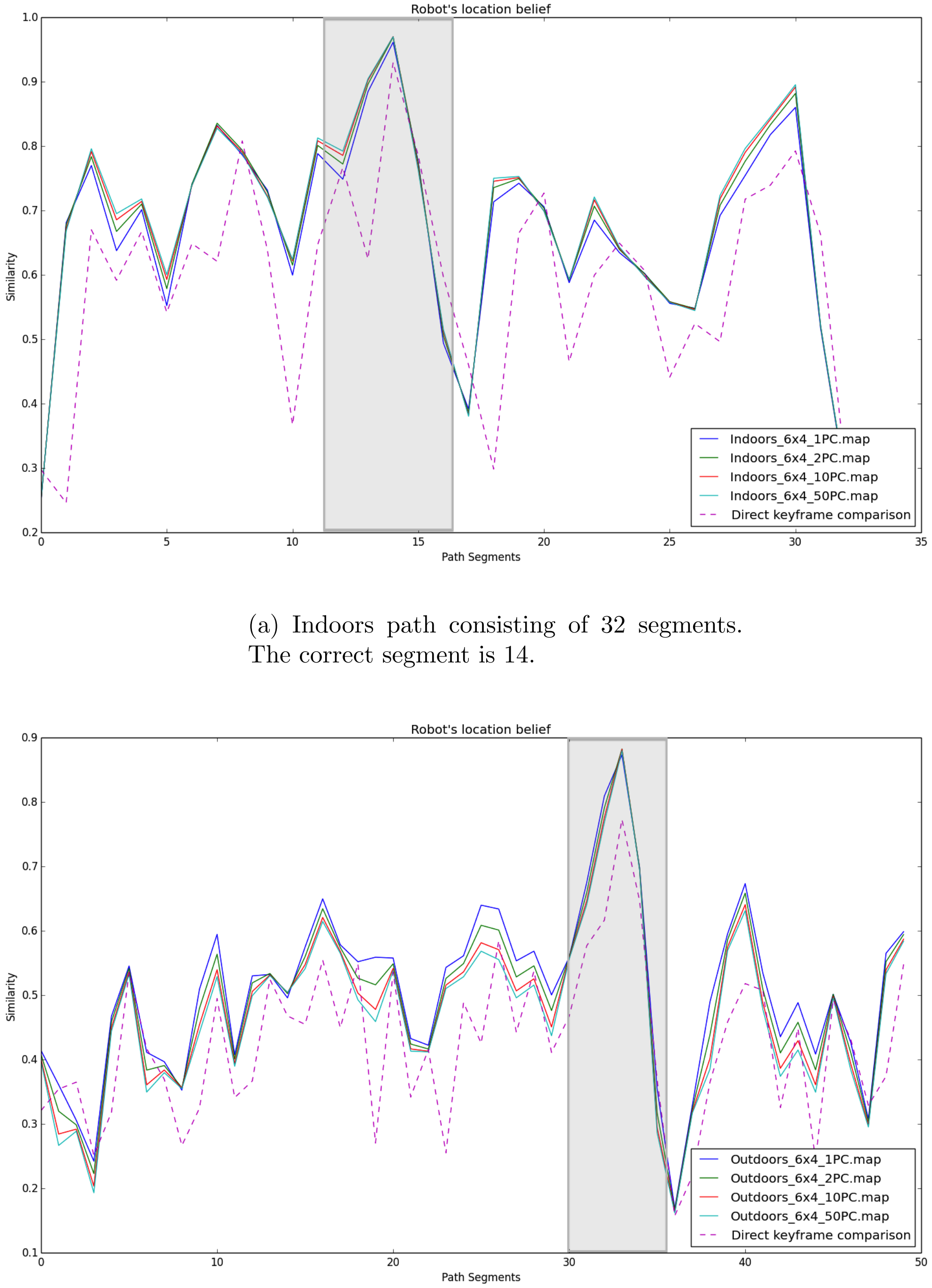


Figure 4.1: Sample images from outdoor's(left) and indoor's(right) datasets.

Initially the *eigensegment* method needed to be verified in a simple and intuitive way. The robots belief about where it is along a path can be visualized as a 2D graph with position on the x-axis and likelihood on the y. The experiment shown in Figure 4.2 illustrates the belief the robot has about where it is along a path given an image. The *eigensegments* are trained using the first loop of the datasets. From the second loop an image near the middle of the loop are selected and compared with all the trained segments. The number of principal components used is an important variable, determining how much of the original descriptor's information must be maintained for correct localization. The path segments are trained with 2, 10, and 50 principal components. Figure 4.2 shows the spike in similarity near the middle segments and how the number of principal components affects the result [1]. Interestingly the number of principal components does not appear to significantly effect the method, 2 PCs giving relatively the same results as 50 PCs.

[1] The spike in similarity is slightly offset from the graphs middle due to the varying size of segments, some segments are larger than others.



(a) Indoors path consisting of 32 segments.
The correct segment is 14.

(b) Outdoors path consisting of 49 segments.
The correct segment is 33.

Figure 4.2: A visual demonstration of the *eigensegment* method localizing a robot along a path. The x-axis represents the path in terms of the sequence of segments, the y-axis the similarity (1-cosine distance) of the chosen image to each *eigensegment*. The image chosen is from the center of the path. (A) Shows the results of an indoor environment and (B) outdoors. The results shown correctly localized the robot the highlighted region shows the local window searched during the robots replay phase to localize the robot. Notice the direct comparison curve for indoor environments is not at as convex as the proposed *eigensegment* method.

The ultimate performance of any localization method is how often it correctly localizes the robot, or the percentage of correct localizations out of a test set. In this experiment 1000 random test images are selected and compared with *eigensegments* containing 2, 10, and 50 PCs. Figure 4.3 shows the results as compared with the previous method of comparing keyframes with the current view. The *eigensegment* method outperforms the keyframe comparisons, achieving 72% vs. 15% correct localizations. Further the number of PCs does not appear to affect performance considerably. Outdoors, a slight positive correlation of correct localizations with the number of PCs can be seen however, the same is not true for indoors.

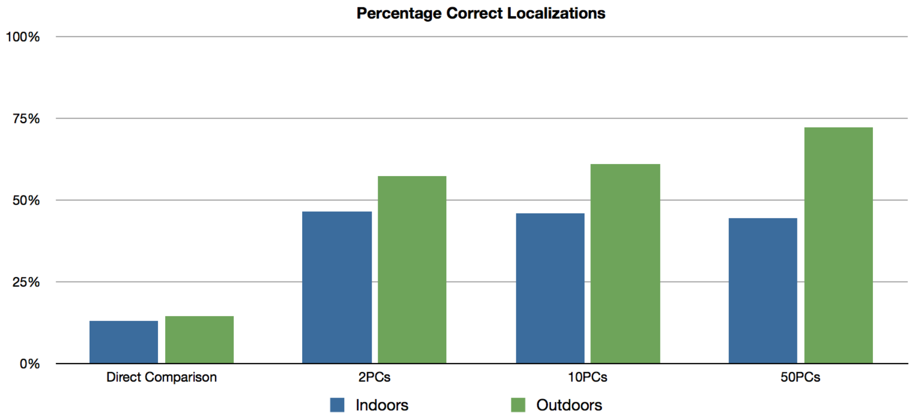


Figure 4.3: The % correct localization of the *eigensegment* method vs. keyframe comparisons. Each bar represents the percentage of correct localizations out of 1000 tests. Three paths are trained each with a different number of PCs namely 2, 10, and 50. Each test randomly selects an image from the second loop of a dataset and compares it to all the *eigensegments* of a path.

These results show improvement over the previous method of keyframe comparisons but is still only 50% effective. During homing we do not perform a global search of all segments to localize the robot. Instead we need only check in a neighbourhood around the current segment. Visual homing allows for this optimization because the the robot proceeds sequentially along the path from one segment to the next. The method maybe only 50% accurate in a global search, within a local search window the performance has proven to be effective for homing. Figure 4.2 shows similarity around the correct values is a convex curve with the correct segment at the apex.

4.3 Homing Experiments

In this section, the visual homing method is experimentally demonstrated. To evaluate this new method of homing, the experiments have been divided into indoor and outdoor environments, each with challenges. Within each environment we present experiments that demonstrate the system is capable of handling specific challenges. Each experiment is performed in real-time with certain changing conditions. Obstacle avoidance has not been considered and the experiments are conducted in quiet conditions. People still appear and pass through the field of view while not obstructing the view. The robot is able to complete the path successfully in these cases.

Initial experiments conducted empirically tuned the parameters of the system. Table 4.1 shows the values for each parameter. Gabor-gist requires convolution of the input image with a filter bank. Smaller images of course allow for faster computation. However heading correction requires a certain amount of image detail to choose the correct heading. This leads to a trade-off in the speed of computation vs. image detail. A final image size of 600x200 from 640x480 was arrived at by repeating paths with different sizes watching how the system responded [2]. The Gabor-gist grid dimensions control encoding of spatial relationships, and has traditionally been set at 4x4. Heading correction however requires a greater amount of spatial detail on the horizontal axis as a result we have chosen a 6x4 grid giving greater spatial resolution. Using the results of section 4.2 we choose two PCs to encode the *eigensegments*. This value has performed well experimentally while minimizing storage requirements. The creation of segments is determined using a threshold of similarity between the latest segment's keyframe and current view and in experiments the value of 0.1 - 0.3 radians was found to produce similar results which proved effective. Given the current segment i only a local window $i-1, i+4$ is searched to localize the robot, the window's range was chosen empirically. Search window size is the size of window used to search the current view for heading correction, a value of 300x200 pixels was selected empirically. Each of these parameter values is held constant throughout the following experiments.

Parameters	**Value**
Image Size	600x200 pixels
Gabor-gist grid	6x4 cells (100x50 pixels/cell)
Eigensegment PCs	2 PCs
Segment threshold	0.1 rad
Localization window	$[i-1, i+4]$
Search window size	300x200 pixels

Table 4.1: The parameters of the system. Image size refers to the size of image used to create the Gabor-gist descriptor. Gabor-gist grid refers to the spatial resolution the descriptor maintains. *Eigensegment* PCs control how much information is retained to characterize a path segment. Localization window controls which segments around the current segment i are checked during segment selection.

During initial experiments replay performance was found to increase when the visual path is encoded by three independent visual paths. During training three separate gist windows offset from each other are trained (see Figure 4.4). Each gist window encodes the path as described in Section 3.1. Each window independently determines when to transition between segments and returns its heading correction signal. During replay, the heading correction signals from each window are averaged together to determine the final correction signal using a weighted average. The weights of each signal are given by the difference between a signals max and min

[2] We also leverage a GPU to convolve the input image with the Gabor filter bank.

similarity over the windowed search. This difference is a measure of how successful the search was in finding the correct image region to steer towards.

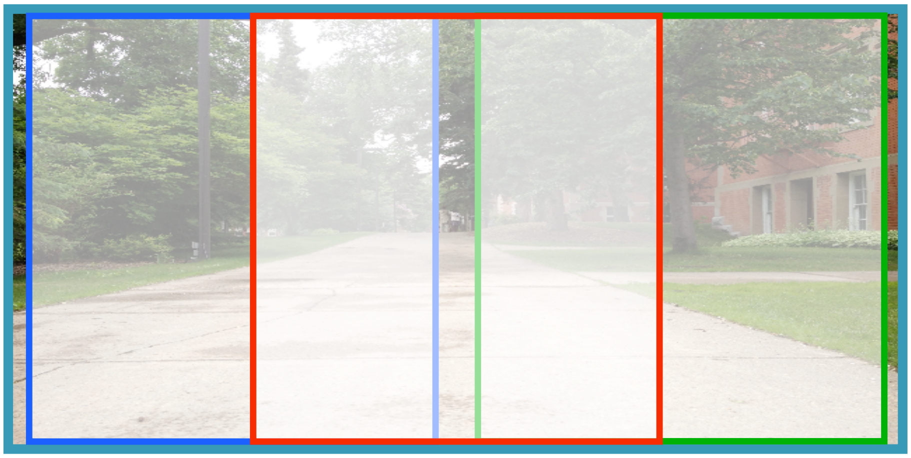
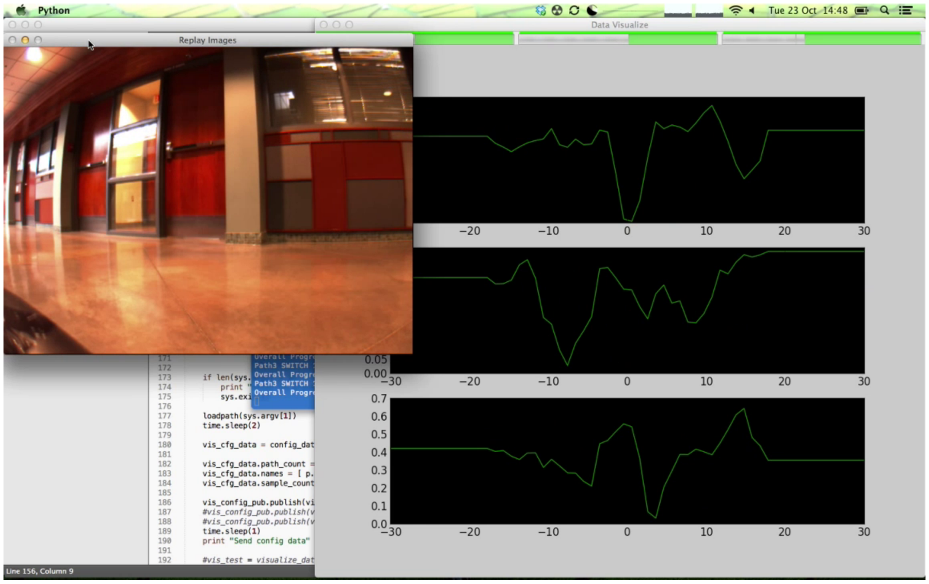


Figure 4.4: Three visual paths blue, red and green offset from each other encode the path using different regions of the image. During replay, each path returns a heading correction signal, which are averaged together to create the final correction signal.

During the early homing experiments the robot would often fail to arrive at the destination, especially if tight turns where required. Initially only a search window from the center of the image was used to make heading corrections. The approach was moderately successful, achieving $\sim 50\%$ success rate. The problem appeared to occur primarily in a turn. In a turn the current view can change quickly and cause both incorrect heading correction and poor localization. Three paths gives better results because if one path is temporarily incorrect the other two generally produce the correct answer. This method raised the success rate to 80% during our experiments.

Figure 4.5 shows a graphical view of the system navigating an indoors path. On the left is the robot's current view I_t. At the top right three progress bars show the progress along the path, while below three similarity graphs show the robot's heading corrections from the three paths (Similarity is in cosine distance and not 1-cosine, making the lowest point of the curve the most similar). See video "Gabor-gist robots view" [3].

[3] Gabor-gist robots view url `http://youtu.be/M87HiO3j2sc`.



```
    if len(sys.
        print "
        sys.exi

    loadpath(sys.argv[1])
    time.sleep(2)

    vis_cfg_data = config_dat

    vis_cfg_data.path_count =
    vis_cfg_data.names = [ p.
    vis_cfg_data.sample_count

    vis_config_pub.publish(vi
    #vis_config_pub.publish(v
    #vis_config_pub.publish(v
    time.sleep(1)
    print "Send config data"

    #vis_test = visualize_dat
```

```
Overall Progr
Path3 SWITCH
Overall Progr
Path3 SWITCH
Overall Progr
```

Figure 4.5: A visualization of a running robot. The top left is the view from the robots camera I_t. On the right is a view of three visual paths being traversed simultaneously. The similarity curves are shown as just cosine distance, thus smaller values are more similar. On the top right are progress bars showing the robots progress along the path.

4.3.1 Indoor Experiments

Indoor environments present a challenge to homing methods because they often have many repeating patterns, which cause visual aliasing. Visual aliasing presents a challenge to both heading correction and localization. Many indoor environments use a repeating pattern, when employing a window search this repetition can cause search regions to look alike, providing no useful information for heading correction. Secondly repeating patterns cause many separate locations indoors to appear similar, complicating localization using images. The locations selected for the indoor experiments contain repeating patterns as well as large blank walls.

The indoor experiments are divided into straight and turning. Straight paths demonstrate the robots ability to traverse a hallway and primarily test the localization of the robot along a path that has many similar visual elements. Figure 4.6 shows a sample of the robot along the path, notice the repeating visual elements on both sides of the hallway. Table 4.2 gives the path and storage details and compares with the entropy based visual homing [8]. This experiment was repeated five times using the same training path. It achieved a success rate of 100%, five out of five trials. See video "Gabor-gist straight hallway test" [4].

[4] Gabor-gist straight hallway test url http://youtu.be/eKfvNECxVzI

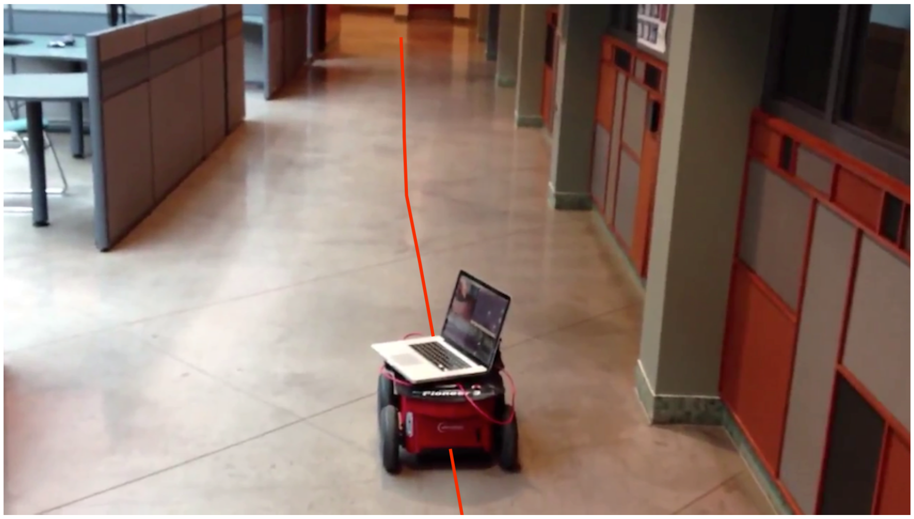


Figure 4.6: Indoor experiment, testing the methods ability to perform proper localization down a hallway. This experiment tests the ability to travel a straight line and also to properly localize the robot in an environment with many repeating patterns. This path is approximately 12 m. The robot completed the path successfully.

Method	Path length	Segments/m	Storage
Gabor-gist	12 m	∼3	648 KB
Entropy	12 m	3	1.3 MB

Table 4.2: Comparison of the storage space and segment density of Gabor-gist homing vs entropy homing in a straight hallway environment.

Turning is generally a difficult task for vision only homing methods. Depending upon the angle of the required turn the resulting heading of the robot may not fall into view at the beginning of a turn. Therefore many segments may be required as intermediate steps. These segments are often small encoding only a few images, making correct localization during replay difficult. Figure 4.7 shows the robot performing the test, along with the route in red. Table 4.3 gives the path and storage details and compares with the entropy based visual homing [8]. This experiment was repeated 5 times using the same training path. It achieved a success rate of 80% or 3 out of 5 trials. During the 3rd and 4th test the robot became lost at different locations for unknown reasons. We were unable to reproduce the errors. See video "Gabor-gist turning test" [5].

[5] Gabor-gist turning test url `http://youtu.be/JayN0vKG62w`.

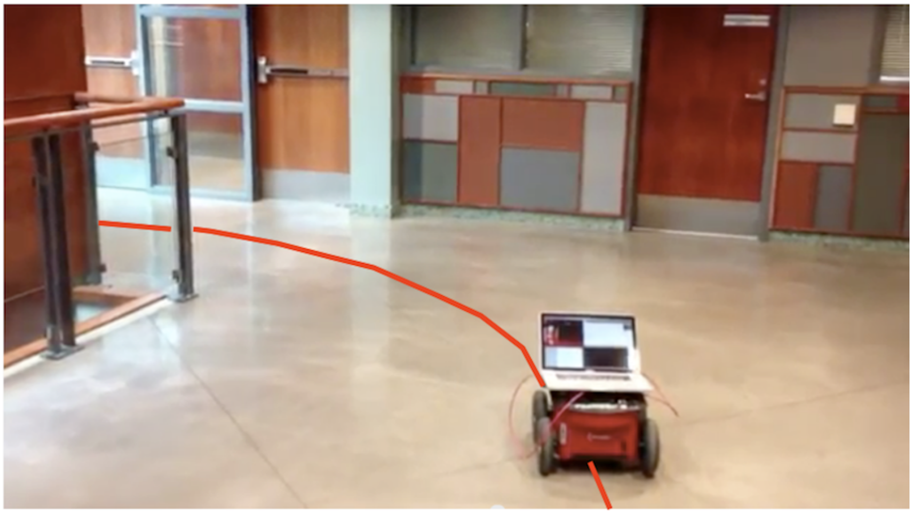


Figure 4.7: Indoor experiment, testing the methods ability to perform the types of turns required for indoor navigation. The path is approximately 10m. The robot completed the path successfully.

Method	Path length	Segments/m	Storage
Gabor-gist	10 m	∼3	560 KB
Entropy	10 m	3	989 KB

Table 4.3: Comparison of the storage space and segment density of Gabor-gist homing vs entropy homing in a indoors turn.

Indoors a robot may be required to work in tight and constrained spaces. The homing method is tested by driving an obstacle type course weaving between chairs. To compare Gabor-gist homing with the entropy method of [8] the segment sampling rate of entropy had to be increased to 5 images/m. Our method required no parameter changes. Figure 4.8 shows the robot navigating the path, while Table 4.4 shows the storage and sample rate details for the experiment. This experiment was repeated 5 times using the same training path. It achieved a success rate of 80% or 4 out of 5 trials. During the 5th test the robot became lost when it failed to localize properly near the end of the path. This was due to not transitioning to the final segment for 2 out of the 3 paths. The path is considered complete when the each is within two segments of the final segment. Here two paths did not achieve this and were giving incorrect heading corrections pulling the robot off course. The robot was however within one meter of the final goal but had to be shutoff manually. See video ”Gabor-gist tight constraint homing” [6].

[6] Gabor-gist tight constraint homing https://www.youtube.com/watch?v=aleNT8_V3Bs.

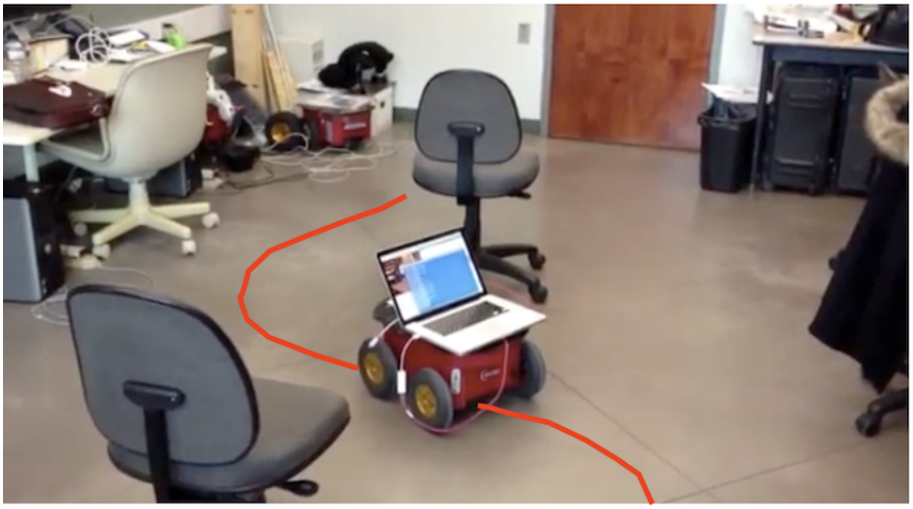


Figure 4.8: Indoor experiment, chosen to test the methods ability to perform in constrained spaces. This path is approximately 5 m.

Method	Path length	Segments/m	Storage
Gabor-gist	5 m	~3	360 KB
Entropy	5 m	5	1.5 MB

Table 4.4: Comparison of the storage space and segment density for Gabor-gist homing vs entropy homing in a tight indoor environment. For the entropy method to navigate the path the segment sampling rate was increased to 5 images/m.

4.3.2 Outdoor Experiments

Outdoor environments provide a different set of challenges to visual homing. Indoors the primary problem is one of visual aliasing, in outdoor locations the primary problem is one of changes in lighting. The outdoor experiments consisted of a 100m route, the first half of which is dominated by trees and shrubs, the second half of two brick buildings. The route was chosen for its variety of visual elements common in outdoor environments. The route was trained using the parameters of Table 4.1 in the afternoon. The route was then repeated later the next day around 10am with different lighting conditions namely overcast. See Figure 4.9 for a sample of the training and replay images. In this figure one can see the differing shadows cast on the ground. Figure 4.10 shows the robot replaying the outdoor path. Table 4.5 gives a comparison between our method and entropy homing in terms of storage space. This experiment was repeated 3 times using the same trained path. The replay phase was done once immediately after training and twice the next day when lighting conditions had changed. A success rate of 100% or 3 out of 3 trials was achieved. When replayed under different lighting conditions the robot would at times visibly deviated from the proper heading mainly at the beginning when the concrete walkway dominates the robot's view. This is due to the shadows present during training but not during replay see Figure 4.9. See video ”Gabor-gist homing outdoors test" [7]. At time 0:00-0:08 the deviation can be seen.

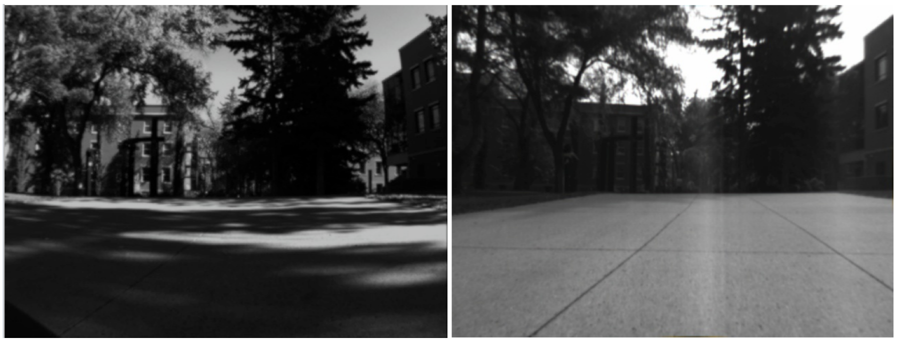


Figure 4.9: Two images one from training (left) taken in the afternoon and one from the replay the next morning (right). The images show a distinct change in lighting conditions particularly on the ground. The robot still completed the path correctly.

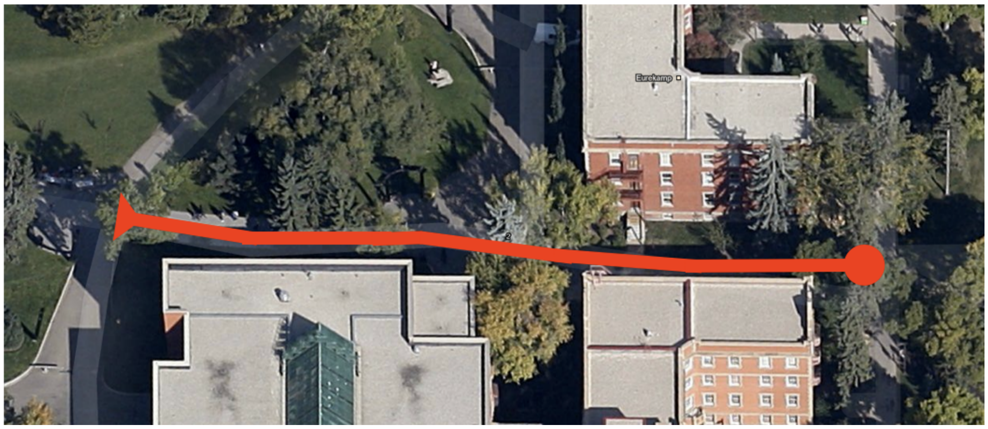


Figure 4.10: The outdoor path of approximately 100 m seen from above.

[7] Gabor-gist homing outdoors test url http://youtu.be/kof6o6_AmV8

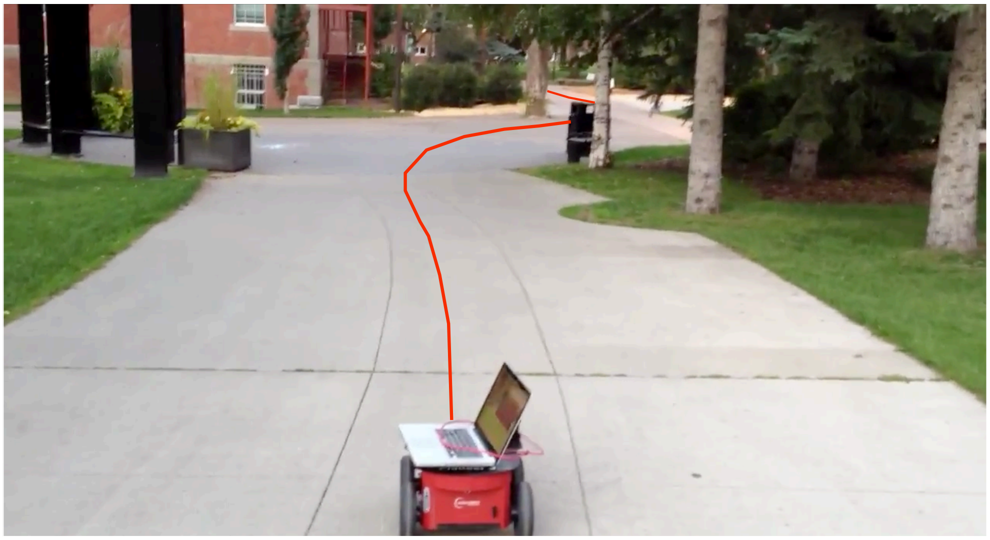


Figure 4.11: The robot replaying the outdoor path using Gabor-gist homing.

Method	Path length	Segments/m	Storage
Gabor-gist	100 m	~0.5	1 MB
Entropy	100 m	3	14 MB

Table 4.5: Comparison of the storage space and segment density for Gabor-gist homing vs entropy homing in an outdoor environment.

4.4 Summary

This section details the experiments used to demonstrate Gabor-gist visual homing. Section 4.2 covered the initial investigations of *eigensegments*. Section 4.3.1 covers indoor experiments while Section 4.3.2 covers the outdoor experiments.

To validate the use of *eigensegments* for localization several simulations are presented in Section 4.2. These simulations first demonstrate the intuition behind the method by graphing a likelihood function of one image against all the segments of a path, showing a distinct spike at the correct segment. The performance of the method is then tested by determining the percentage of correct localizations from 1000 random images. These results are compared with another method of directly comparing keyframes with the current view. The *eigensegment* method achieves a performance of 72% vs. 15%.

The system detailed in this thesis is not without failure cases. Gabor-gist homing often fails during turns. This is primarily due to insufficient overlap in images, where the current view does not contain enough of the goal segment's keyframe to make proper heading corrections. During training a similarity threshold is used for keyframe selection, this a constant threshold and can choose keyframes to far apart, during replay this results in not being able to see the goal image causing incorrect heading correction. Another problem in turns is localizing the robot. During a turn the current view changes a great deal in a short time. In training this can lead to insufficient training images to represent a segment. In replay it then becomes difficult to properly localize the robot and make proper heading corrections. The cause of this can be due to an improper similarity threshold during keyframe selection in training, or due to an incorrect localization. Tight turns are often also prone to incorrect localization because in a turn the current view changes quickly and as a result a segment does not enough training images.

Section 4.3.1 demonstrates the results of testing the algorithm in an indoor environment. Several situations are tested, including straight, turns, and constrained environments. Section 4.3.2 covers the outdoor experiments where the method is tested outdoors where lighting changes in terms of shadows at different times of day. Each of these tests completed successfully and required a fraction of the storage space required by previous methods.

Chapter 5

Conclusions and Future Directions

5.1 Conclusions

Vision-based robotics is an active area of research with significant progress being made in developing both visual navigation algorithms during the past three decades. The interest in using cameras for sensing comes from the observation that images provide a natural way of perceiving the environment. Most current research efforts in visual homing tend to focus on 3D reconstructions of the environment to develop controllers taking some type of explicit models of the robot, cameras, or environments into account. Many of these systems however require large amounts of computation and special sensors beyond a camera. There is a push to develop algorithms that can be applied to hardware constrained systems.

Because robots need to act on the fly, computer vision algorithms should run in real-time. There are challenges in real-time processing of visual information. Most notably, visual measurement is often corrupted by noise and outliers. Uncertainties in the motor actuation always exist as well. This imposes significant challenges on algorithm design. To perform practical homing in unstructured settings vision-based robots require algorithms which (1) do not depend on explicit models, and (2) are robust against sensing uncertainties and outliers. Consequently, this thesis has concentrated on the development of a visual homing algorithm to avoid the requirements of explicit models. This thesis is a step towards answering the question on how to efficiently control the motion of a vision-based robot on occlusion-free paths in different environments.

In the contributions we have emphasized the importance of using the biologically inspired gist image descriptor which compactly represents an image, while maintaining the information crucial to vision-based homing. We have described how to use the gist descriptor for heading correction using a simple window search across the horizontal axis of the current image. Further, we have introduced a novel method for encoding a segment of the visual path, which we have termed an *eigensegment*. Using PCA to encode the statistically important qualities from a set of consecutive images gist descriptors, we demonstrate that it performs better then keyframe comparisons. Although we have presented only the first steps towards a practical implementation, hopefully we have shown the potential of using Gabor-gist in the area of visual homing.

5.2 Future Directions

The method presented is a novel approach to visual homing. The goal of this research is to employ compact visual homing using Gabor-gist. Future research could include using Gabor-gist to create path templates, useful beyond on explicitly trained path. The Sparse Distributed Memory mentioned in the Chapter 2 could also benefit from the Gabor-gist compact image representation. Gabor-gist itself is a broad and general approach to scene classification many, texture analysis Gabor filters are tuned to detect specific patterns this could be applied here.

During an indoor experiment the robot made a mistake but exhibited the ability to use a trained path to correctly navigate in a location it had never before seen. The experiment was repeated twice more with the same result, suggesting that Gabor-gist might provide a way to encode a general path like structure that could be applied to other never before seen paths. In future work it would be of interest to train a path and then place the robot at a different but visually similar location and see if it performs the proper actions. This might prove a useful way of making path templates and associated actions that could be used to navigate a path the robot has not been explicitly trained for. This is similar to the work of Ackerman [1].

In Chapter 2 we introduced the use of a sparse distributed memory [31], the main limitation of which is in the storage requirements of 0.1 bits per bit of traditional memory. In its current setup the authors encode a pixel image of 80x64. Gabor-gist being a compact representation of an image could alleviate the storage problems considerably from needing to encode 5120 (80x64) pixels/keyframe to 480 (6x4x20) floats/keyframe.

Lastly Gabor-gist uses a even distribution of filters across the scale and frequency ranges. Others have tuned Gabor filters to detect certain objects or patterns [49, 9, 30]. It would be interesting to tune the filter bank of Gabor-gist in such a way so as to increase the performance of the heading correction or localization. Heading correction performance may be increased by tuning the filters towards more vertical edges. Localization might benefit from more lower frequency filters rather than noisy high frequency ones. It is also conceivable that separate filter banks could be used depending on the environment, one bank for outdoors and another indoors.

Bibliography

[1] C Ackerman and L Itti. Robot steering with spectral image information. In *IEEE Transactions on Robotics*, volume 21, pages 247–251. IEEE, 2005. 5, 35

[2] J L Aguilar. An adaptive sparse distributed memory. In *Proceedings of the International Joint Conference on Neural Networks 2003*, volume 3, 2003. 5

[3] R Basri, E Rivlin, and I Shimshoni. Visual homing: surfing on the epipoles. *International Journal of Computer Vision 33.2*, 33(2):117–137, 1999. 4, 5, 11

[4] H Bay, A Ess, T Tuytelaars, and L Vangool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. 4

[5] Zhichao Chen and S T Birchfield. Qualitative Vision-Based Path Following. *IEEE Transactions on Robotics*, 25(3):749–754, 2009. 4, 5, 11

[6] Y Cheng, M Maimone, and L Matthies. Visual odometry on the Mars Exploration Rovers. In *2005 IEEE International Conference on Systems Man and Cybernetics*, volume 1, pages 903–910. Ieee, 2005. 2

[7] Amaury Dame and Eric Marchand. Entropy-based visual servoing. *2009 IEEE International Conference on Robotics and Automation*, pages 707–713, 2009. 6

[8] Amaury Dame and Eric Marchand. A new information theoretic approach for appearance-based navigation of non-holonomic vehicle. *2011 IEEE International Conference on Robotics and Automation*, pages 2459–2464, 2011. 6, 11, 14, 27, 28, 29

[9] J G Daugman. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics Speech and Signal Processing*, 36(7):1169–1179, 1988. 10, 35

[10] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of GIST descriptors for web-scale image search. *Proceeding of the ACM International Conference on Image and Video Retrieval CIVR 09*, page 1, 2009. 7

[11] P Gaussier, C Joulain, S Zrehen, J P Banquet, and A Revel. Visual navigation in an open environment without map. *Proceedings of the 1997 IEEERSJ International Conference on Intelligent Robot and Systems Innovative Robotics for RealWorld Applications IROS 97*, 2:545–550, 1997. 5

[12] Sennay Ghebreab, H S Scholte, V A F Lamme, and Arnold W M Smeulders. A Biologically Plausible Model for Rapid Natural Scene Identification. *Advances in Neural Information Processing Systems*, pages 629–637, 2009. 7

[13] C Giovannangeli, P Gaussier, and G Desilles. Robust Mapless Outdoor Vision-Based Navigation. *2006 IEEERSJ International Conference on Intelligent Robots and Systems*, 1(1):3293–3300, 2006. 5

[14] Michelle R Greene. *A Global Framework for Scene Gist*. PhD thesis, Massachusetts Institute of Technology, 2009. 7

[15] J J Guerrero, D Kragic, and P Jensfelt. Switching visual control based on epipoles for mobile robots. *Robotics and Autonomous Systems*, 56(7):592–603, 2008. 11

[16] Yina Han and Guizhong Liu Guizhong Liu. A Hierarchical GIST Model Embedding Multiple Biological Feasibilities for Scene Classification. *Pattern Recognition ICPR 2010 20th International Conference on*, 0:3109–3112, 2010. 7

[17] Blakeslee Sandra Hawkins Jeff. *On Intelligence*. Times Books, 2004. 5, 11

[18] Ian Horswill. Polly : A Vision-Based Articial Agent. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1993. 5

[19] Aiweng Jiang, Chunheng Wang, Baihua Xiao, and Ruwei Dai. A New Biologically Inspired Feature for Scene Image Classification. In *Pattern Recognition ICPR 2010 20th International Conference on*, volume 0, pages 758–761. Ieee, 2010. 7

[20] J P Jones and L A Palmer. An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6):1233–58, 1987. 9

[21] S D Jones, C Andresen, and J L Crowley. Appearance based process for visual navigation. In *Proceedings of the 1997 IEEERSJ International Conference on Intelligent Robot and Systems Innovative Robotics for RealWorld Applications IROS 97*, volume 2, pages 551–557. Ieee, 1997. 5

[22] Pentti Kanerva. Sparse distributed memory. *IEEE Transactions on Neural Networks*, 18(3):333–335, 1989. 5

[23] J Kosecka, Liang Zhou Liang Zhou, P Barber, and Z Duric. Qualitative image based localization in indoors environments. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2003 Proceedings*, volume 2, pages II–3–II–8. IEEE Comput. Soc, 2003. 5

[24] B J A Kröse, N Vlassis, R Bunschoten, and Y Motomura. A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, 19(6):381–391, 2001. 5

[25] Yang Liu and Hong Zhang. Visual loop closure detection with a compact image descriptor. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1051–1056, October 2012. 10

[26] D G Lowe. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2([8):1150–1157 vol.2, 1999. 4

[27] Edelman G M and Finkel L H. Neuronal group selection in the cerebral cortex. *Dynamic Aspects of Neocortical Function*, 1984. 9

[28] Y Matsumoto, K Sakai, M Inaba, and H Inoue. View-based approach to robot navigation. *Proceedings 2000 IEEERSJ International Conference on Intelligent Robots and Systems IROS 2000*, 3(5):1702–1708, 2000. 5

[29] Yoshio Matsumoto, Masayuki Inaba, and Hirochika Inoue. Visual navigation using view-sequenced route representation. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 1, pages 83–88. IEEE, 1996. 1, 3, 4, 5, 6, 14

[30] R Mehrotra, K R Namuduri, and N Ranganathan. Gabor filter-based edge detection. *Pattern Recognition*, 25(12):1479–1494, 1992. 10, 35

[31] M Mendes, M Crisostomo, and A P Coimbra. Robot navigation using a sparse distributed memory. In *2008 IEEE International Conference on Robotics and Automation*, volume advance on, pages 53–58. Ieee, 2008. 5, 14, 35

[32] A. C. Murillo, P. Campos, J. Kosecka, and J. J. Guerrero. Gist vocabularies in omnidirectional images for appearance based mapping and localization. *Murillo, A. C., et al. "Gist vocabularies in omnidirectional images for appearance based mapping and localization." 10th IEEE workshop on omnidirectional vision, camera networks and non-classical cameras,(OMNIVIS), held with robotics, science and systems.*, 2010. 7

[33] S K Nayar, H Murase, and S A Nene. Learning, positioning, and tracking visual appearance. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 3237–3244. IEEE Comput. Soc. Press, 1994. 5

[34] A Oliva and P G Schyns. Diagnostic colors mediate scene recognition. *Cognitive Psychology*, 41(2):176–210, 2000. 7

[35] Aude Oliva, Women Hospital, and Longwood Ave. Modeling the Shape of the Scene : A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001. 6, 7

[36] Josien P W Pluim, J B Antoine Maintz, and Max A Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, 22(8):986–1004, 2003. 6

[37] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. *IEEE Conference on Computer Vision and Pattern Recognition (2009)*, (July):413–420, 2009. 7

[38] C. Sagues and J.J. Guerrero. Visual correction for mobile robot homing. 50(1):41–49, 2005. 5

[39] Atsushi Sakai, Yuya Tamura, and Yoji Kuroda. Visual Odometry Using Feature Point and Ground Plane for Urban Environments. *Journal of Field Robotics*, pages 654–661, 2010. 2

[40] C Siagian and L Itti. Mobile robot vision navigation & localization using Gist and Saliency. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4147–4154, October 2010. 4, 5, 7, 10

[41] Christian Siagian and Laurent Itti. Rapid Biologically-Inspired Scene Classification Using Features Shared with Visual Attention. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):300–312, 2007. 7

[42] R Sim and G Dudek. Learning environmental features for pose estimation. *Image and Vision Computing*, 19(11):733–739, September 2001. 5

[43] Niko Sunderhauf and Peter Protzel. BRIEF-Gist - Closing the loop by simple means. In *2011 IEEERSJ International Conference on Intelligent Robots and Systems*, pages 1234–1241. Department of Electrical Engineering and Information Technology, Chemnitz University of Technology, 09111, Germany, IEEE, 2011. 7

[44] A Torralba and A Oliva. Depth estimation from image structure. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 24, pages 1226–1238. IEEE Computer Society, 2002. 7, 10

[45] Antonio Torralba, Kevin P Murphy, William T Freeman, and Mark A Rubin. Context-based vision system for place and object recognition. *Proceedings Ninth IEEE International Conference on Computer Vision*, 1(March):273–280 vol.1, 2003. 1, 7, 10

[46] I Ulrich and I Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings 2000 ICRA Millennium Conference IEEE International Conference on Robotics and Automation Symposia Proceedings*, volume 2, pages 1023–1029. Ieee, 2000. 5

[47] Willow Garage. ROS (Robotic Operating System). 21

[48] Jurgen Wolf, Wolfram Burgard, and Hans Burkhardt. Using an Image Retrieval System for Vision-Based Mobile Robot Localization. *Image and Video Retrieval*, pages 108–119, 2002. 5

[49] Dengsheng Zhang, Aylwin Wong, Maria Indrawan, and Guojun Lu. Content-based Image Retrieval Using Gabor Texture Features. *Image Rochester NY*, 3656 LNCS:13–15, 2000. 35

[50] Chao Zhou, Yucheng Wei, and Tieniu Tan. Mobile robot self-localization based on global visual appearance features. In *2003 IEEE International Conference on Robotics and Automation*, volume 1, pages 1271–1276. Citeseer, 2003. 5