Geometric hashing for camera based localization

Tan, Wei Chian

2012

Tan, W. C. (2012). Geometric hashing for camera based localization. Master's thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/49984

https://doi.org/10.32657/10356/49984

GEOMETRIC HASHING FOR CAMERA BASED LOCALIZATION

TAN WEI CHIAN

SCHOOL OF COMPUTER ENGINEERING

2011/2012

Acknowledgement

I would like to take this opportunity to extend deepest gratitude to my supervisor, Associate Professor Cham Tat Jen for his guidance and encouragement throughout the project. I would also like to thank my senior, Dr. Pham Minh Tri and my colleague, Mr. Arridhana Ciptadi in the team for brainstorming sessions, discussions and ideas.

Last but not least, my heartfelt thanks would also go out to laboratory executives and my friends in Centre for Multimedia and Network Technology (CeMNet) of School of Computer Engineering for numerous discussions and strong supports along the way.

This project would not have been possible without their experiences, ideas, patience, inspirations and love.

Thank You.

Wei Chian

Table of Contents

Abstract

The most popular localization system now, Global Positioning System (GPS), is well known for its limitation in high rise urban areas due to difficulty in establishing line of sight to multiple GPS satellites. A vision or camera based localization system is an interesting alternative to consider for localization.

A camera based localization system has been established previously [1], assuming that the only available prior information is a two-dimensional (2D) plan view of a city region. Given a query image taken in the same city region, the basic approach of localization is to establish correspondence between the query image and the 2D map, based on a new feature called Vertical Corner Line Hypothesis (VCLH), hypothesis of a vertical building corner in an image. A VCLH is characterized by position of the vertical line or building corner in the image, and orientations of the neighbouring plane normal. The set of VCLHs extracted from the input image is called VCLH signature. Matching is performed by identifying the camera position on the 2D map with the closest VCLH signature to the input image one using Random Sample Consensus (RANSAC) [2]. However, the search for best camera location is computationally expensive. Hence, this project aims to develop a speedup framework to solve the problem. Geometric Hashing [3], hashing based on geometric information such as keypoints invariant to translation and rotation, is employed in this project.

A Geometric Hashing system based on VCLH is developed. The system is similar to the original framework. During pre-processing, VCLH signature obtained from each camera pose in the 2D map is quantized into bins and inserted into a two dimensional (2D) hash table. During online retrieval, the query signature is quantized and a voting mechanism is used to identify good candidates (camera pose) for further verification. Experiments show that significant speed gains have been obtained through the system, from direct search of ten minutes to removal of large pool of candidates within ten seconds in our Matlab [4] implementation. However, the performance is not good, none of the shortlists returned for each query location include the ground truth, both with and without Geometric Hashing. This is mainly due to poor detection results of VCLH feature.

Next, the concepts of connectivity and omnidirectional views are introduced to the system and incorporated into a feature known as Structural Fragments (SF). Connectivity refers to a plane facade between two VCLHs and corresponds to a straight line in 2D plan view. An SF is characterized by a set of VCLHs (points in 2D plan view) and relevant connectivity information. During pre-processing, outline of each building in the 2D map is quantized into bins by taking any two building corners as the basis to establish a coordinate system. The process is repeated for all possible pairs of basis. During online retrieval, given an SF, similar coordinate system establishment procedures are taken and a voting mechanism is used to retrieve close candidates. This avoids the need of comparing the query SF to each possible SF in the 2D map (linear to sublinear search improvement). In addition, SF provides further speed gain because the framework no longer requires dense sampling of camera viewpoints

during pre-processing. Experiments show a significant improvement on accuracy, with much greater speed gains. Uniqueness test was also carried out to investigate how discriminative the signatures are theoretically. Experiments reveal that SF signatures are surprisingly unique.

In short, the aim of speeding up has been achieved. The Geometric Hashing system developed gives good performance. However, there are still some problems with both VCLH and SF detection. Next step could be looking into improvement of feature detection or more advanced speeding up techniques such as Locality Sensitive Hashing (LSH) [5] and Randomized Tree [6] or Forest [7] for better performance.

List of Figures

List of Tables

Chapter 1 Introduction

Humans are able to locate themselves based on what they see, but only in locations that they are already familiar with. Different tools have been used to extend the ability, ranging from maps to electronic aids.

The most popular localization method currently, Global Positioning System (GPS), can provide absolute position of the query almost in real time. Ranging from commercial mobile phone applications and commercial flight navigation system to time critical military missions, GPS has proven to be very useful. However, the performance of current GPS systems is often limited by line of sight to satellites, which is difficult in high rise urban areas.

Vision or camera based localization systems can be very interesting. Given a query image captured and some kind of prior information, the system is able to tell the user where the query image is taken. This has the advantage of not having to depend on GPS. It might just be a program that can be installed on user's smartphone and everything can be done locally.

It can be very hard for tourists to locate themselves in a foreign city especially in non-tourist areas, such as areas with similar buildings in terms of geometry and appearance. The system comes in as a great help in this scenario.

Another possible application of this system would be allowing robot soldiers in urban areas to locate themselves easily by a camera mounted on their bodies. This is very useful as this allows the robot soldiers to do more such as collaborating with others and react in a team instead of individual. This would definitely improve the chances of successful operations by the soldiers such as in a mission to rush to and gain control of a vital spot where real time self localization is crucial. This type of system can help to realize the ultimate goal of having

fully automated robot soldiers in the battlefield to replace humans in dangerous missions and hence reduce casualties of the armed forces.

However, a high performance vision based localization system is not easy to achieve. One possible solution to this is to base the localization system on geometrical information. The prior information available is only a two-dimensional plan view of an urban area. This can be easily obtained from satellite imagery. The geometrical information extracted from the image can be used to identify a point (camera location and viewing angle) where it has the closest view to the query image.

One big advantage of this design is that neither three-dimensional nor appearance information of the interest region is needed. This greatly reduces the computing resources required as three-dimensional models demand intensive computing resources. In a military context, this has a great advantage over 3D modelling approach, as 2D data of enemy territories can be much more rapidly acquired and refreshed.

1.1 Objectives and Scopes

A camera based localization framework has been established based on the method discussed above. More details will be provided in the first part of Chapter 3. However, the system is quite slow as it involves a linear search of all the possible hypotheses in the 2D map during searching for the best location and viewing angle. This project aims to develop a fast framework to address this problem. The basic approach is to eliminate a big pool of hypotheses, and only a small portion of the candidates are passed to the verification stage for detailed matching. The best candidate is reported to the user subsequently.

1.2 Report Overview

This report consists of five chapters. Chapter 1 is an introduction to the general problem of vision or camera based localization and establishes a problem statement for this project. Chapter 2 provides review of relevant works in the computer vision community. Relevant speedup frameworks are divided into different groups and detailed discussion is provided in each section.

Chapter 3 provides information about methods employed in the system. The chapter starts with some details about the already developed system framework, and followed by details of the speedup method proposed. Chapter 4 presents details about extension of the current system to connectivity information (plane facade) and multi views for more discriminative power. A new feature called Structural Fragment (SF) is introduced. Another Geometric Hashing framework based on SF is developed. This framework which does not require dense sampling at pre-processing stage provides further discriminative power and speed gain. The chapter starts with details about extension of the already developed system framework to SF, and followed by details of the Geometric Hashing framework proposed.

Chapter 5 provides experimental information of the speeding up frameworks developed. Analysis and detailed discussion are provided subsequently. Chapter 6 is a conclusion chapter. Possible plans for future works are discussed here.

Chapter 2 Literature Review

Previous chapter has presented an introduction to camera based localization and established the project aim. This chapter will discuss relevant works in the computer vision community. The methods are divided into two major categories in general, namely hash based and tree based methods. Detailed discussion of each category is provided.

2.1 Hash Based Methods

Hashing is a very useful technique to conduct search in sub-linear time. Many works in the computer vision community has extended hashing to solve relevant vision problems. Two major directions are reviewed in this section, namely Geometric Hashing and Locality Sensitive Hashing (LSH).

2.1.1 Geometric Hashing

Geometric Hashing introduced by Lamdan and Wolfson [3] has been a very popular technique in computer vision. Given *n* interest points extracted from a model, two are taken as basis to define a coordinate system and the rest are taken as support to the basis and the existence of the model object. This process is repeated for all possible pairs of basis from the n interest points, basically ${}^{n}C_{2}$ times. A hash table is created during this offline stage.

During online recognition, assuming *m* interest points are detected, two are taken as basis and the rest are taken as query to the hash table and *m-2* hashing processes happen. Basically, all records in the hash bin the query points to gets a vote. After the voting stage, records with number of votes higher than a user-defined threshold is passed to the verification stage for detailed matching. Figure 2.1 below provides an overview of Geometric Hashing.

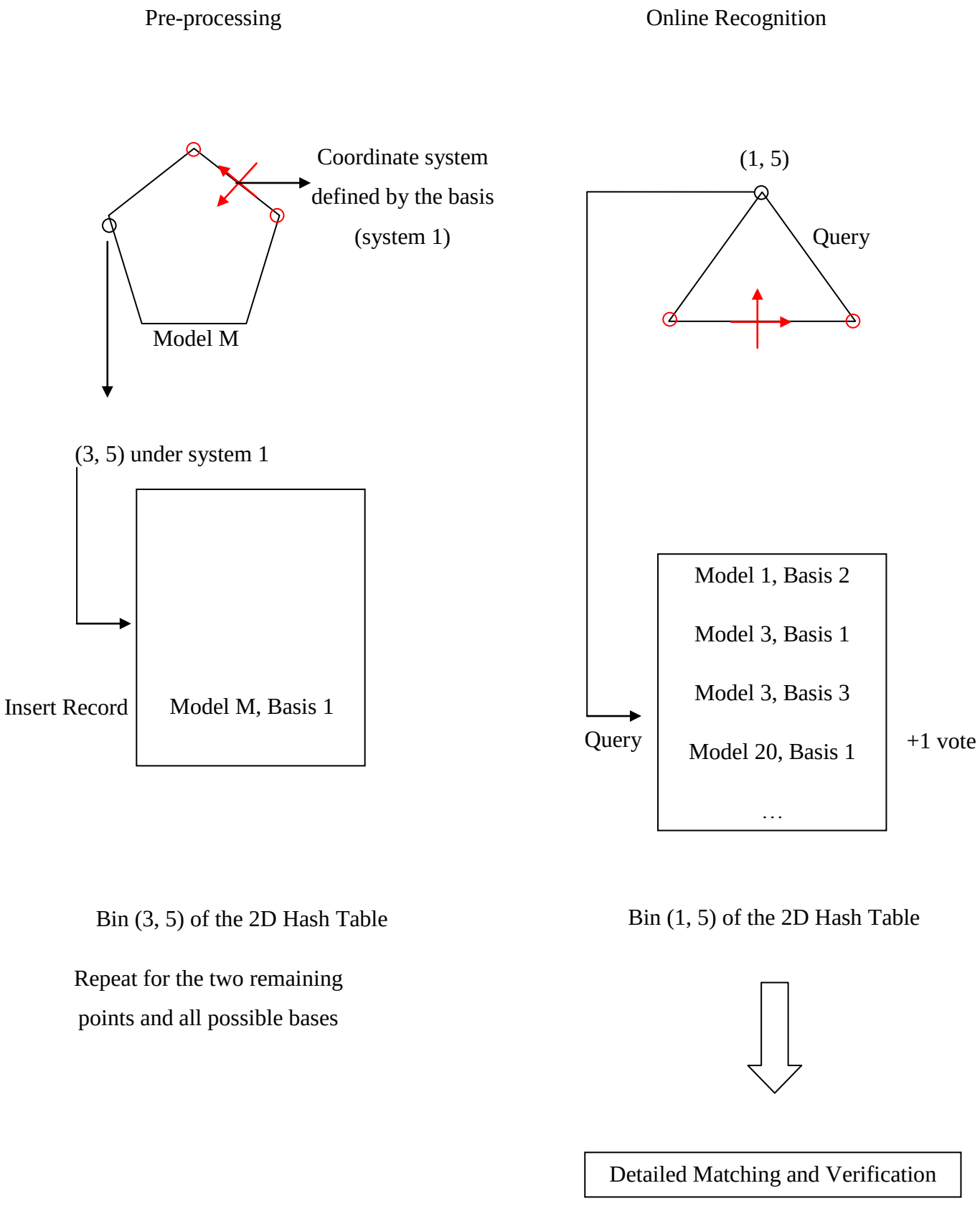


Figure 2.1: Overview of Geometric Hashing

This can save a lot of query time as it avoids an exhaustive search against all the models in the database. The algorithm complexity of the online recognition stage is $O(HS^{c+1})$ where S is the number of features the query contains, H is the time complexity of processing a hash bin and c is the number of features needed to form a basis. Similar to conventional hashing techniques, H depends on distribution of the hash keys. H is $O(1)$ if it is a uniform distribution. The worst case is linear to the number of models. However, the worst case can always be avoided as the hash table is obtained offline, necessary adjustments can be made to obtain a better distribution of the hash keys.

Advantages of Geometric Hashing are that it is able to handle overlapping objects and partial occlusion and is invariant to basis-determined geometric transformations. Recent works in Geometric Hashing tends to focus on geometric invariants under perspective projection and use them as the basic units for Geometric Hashing.

Many subsequent works extended Geometric Hashing to many different features, some of them exploit geometric invariant information to address transformation problems. Significant ones include line features by Tsai [8], an efficient representation of 3D model by 2D projections by Gavrila and Groen [9], sets of ordered triples by Wu and Chang [10] for object recognition, curve by Gueziec and Ayache [11], 3D low level features by Dijck and Heijden [12], generalized CSS method [13] by Mokhtarian, Khalili and Yuen [14], oriented points by Johnson and Hebert [15].

Bebis, Georgiopoulos and Lobo proposed a learning scheme to achieve more uniform distribution in the hash space by Self-Organizing Maps (SOM) [16]. Basically the idea is to make use of the ability of SOM to capture the distribution of the input data to distribute the hash bins over the invariants in a more uniform way. More hash bins will be allocated for denser regions and vice versa. This can solve the problem of the original Geometric Hashing framework where most hash keys fall into few bins only.

Rigoutsos and Hummel [17] proposed a method to view Geometric Hashing as a Bayesian Maximum-Likelihood Framework. Less weight is given if the hash key is more distance away from the hash entry and vice versa. This solves the well known problem of quantization of the hash keys.

More recent works include work by Chum and Matas which extended Geometric Hashing to a new feature called Local Affine Frames (LAF) [18] for matching. Geometric min-Hashing [19] which is based on Min-Hash [20], a technique in near document identification, is applied to Near Duplicate Image Detection (NDID). Spatial information of image features is exploited in the method. It works well with significant occlusion and small overlap of viewing fields.

2.1.2 Locality Sensitive Hashing (LSH)

A more recent evolution of Near Neighbour (NN) Search, Locality Sensitive Hashing (LSH) [5] is proposed for searching of noisy patterns. Natures of the hash functions make the probability of having collision large when two hash values are close enough, typically radius r, and vice versa. Popular hash functions used include Hamming distance [5] and l_1 distance [21]. This property is useful in retrieving seen or learnt patterns given a noisy input pattern. LSH has been widely used in the Computer Vision community lately [22-27]. Some works are reviewed in this section.

Yang, Ooi and Sun [28] proposed to search for similar videos in large video databases and pointed out the well known problem with conventional hashing and hence LSH, namely the non-uniform distribution of the hash keys. Two ways are proposed as solutions. The first is hierarchical: if number of hash keys inside a bin is higher than a threshold, the system will repartition the bins and rehash. The second is called non-uniform partitioning: division of the feature space is no longer uniform but differs from dimension to dimension, depending on distribution of values in that dimension. Dimension with densely distributed values should be chosen with a lower probability and vice versa.

Wang, Tang and Shum [29] proposed a learning based image super resolution and an adaptive locality sensitive hashing algorithm for similarity search in large training data set. A hierarchy of hash tables is built with different widths of projection and adaptive search is used in the query stage. This solves the problem of choosing the best width of projection, where too small a width leads to having no neighbours and too large a width leads to too many neighbours being retrieved.

Recently, Shakhnarovich, Viola and Darrell extended LSH to Parameter Sensitive Hashing (PSH) [30] to solve the parameter estimation problem. A new learning algorithm is proposed to learn a set of hash functions which are optimally relevant to a particular estimation task. In other words, the distance measure now occurs at parameter space instead of input space which is the case for LSH. This solves complex and high-dimensional problems such as pose estimation as everything happens in parameter space now.

Jain, Kulis and Grauman [31] learnt a Mahalanobis distance function that capture relationships between the images well and introduced a method to incorporate the function into LSH. An implicit formulation that enables information-theoretic learning with high-dimensional inputs is also derived. This solves the well known problem with high-dimensional inputs.

2.2 Tree Based Methods

Tree-based methods have been popular in the computer vision community for speeding up search. Sub-linear searching performance can be achieved, depending on some optimal adjustments such as sorting. Some relevant works will be discussed below.

Nister and Stewenius [32] introduced a vocabulary tree structure for efficient large scale object recognition. Similar to recent popular techniques, the method is based on indexing descriptors extracted from image local regions. Local image region descriptors are

quantized hierarchically in a vocabulary tree. One advantage of this method is that the vocabulary tree structure defines the quantization process.

Schindler, Brown and Szeliski [33] used vocabulary tree to solve location recognition problem and introduced a generalized version of vocabulary tree by increasing branching factor of a fixed vocabulary tree effectively for good performance.

Lepetit, Lagger and Fua [6] proposed to treat wide baseline matching of feature points as a classification problem with randomized tree used as classification technique. Use of randomized trees gives a principled way to match keypoints and to choose the most recognizable objects during training stage.

Mikolajczyk and Uemura [34] proposed to use a vocabulary forest of local motion appearance features for action recognition. Large numbers of features with corresponding motion vectors are represented by many vocabulary trees. The large number of trees make the recognition process efficient and robust.

Moosmann, Nowak and Jurie [7] introduced Extremely Randomized Clustering (ERC) Forests for more accurate, robust and faster image patch quantization. It is shown that ERC Forests are very good at learning distance functions of unseen objects, and able to group features that belong to same object and vice versa.

Chapter Summary

Various speeding up methods and their developments in recent years have been reviewed in this chapter. Next, the overall system architecture and methodology for the speeding up framework are presented.

Chapter 3 Geometric Hashing System

Previous chapters have presented an introduction to camera based localization, established problem statement of the project and review of relevant works in the computer vision community. This chapter will discuss the speeding up framework proposed. Overall system architecture will be discussed in brief at first, followed by detailed discussion of the speeding up framework developed.

3.1 Current System Architecture

The overall system is depicted in Figure 3.1. The system takes a query image as input and pre-processes it. Specific features are extracted from the query image and used for matching in the two-dimensional map. The final output of the system is the camera basis with the closest view to the query image. More details will be provided in subsequent sections.

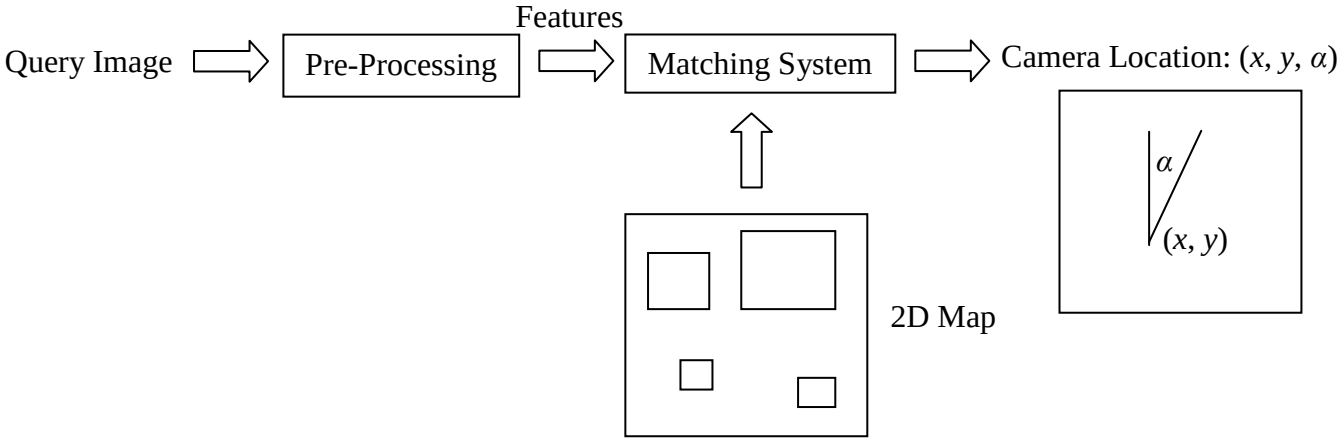


Figure 3.1: Overall Camera Based Localization Framework

3.1.1 Assumptions

We agree that the locality to which we are applying our method is adequately modelled as a Quasi-Manhattan World. This is a relaxation of the more tightly constrained Manhattan World assumption [35] used in other methods, and basically means building facades are piecewise planar and the facade orientations are either parallel or perpendicular to a vertical axis. However, we impose an additional assumption that buildings are also well modelled on a vertical extrusion of a 2D cross section that lies in the ground plane. A typical input image which fits these assumptions is provided in Figure 3.2. Additional considerations on special architectures such as landmark buildings would be required if the assumption is removed later. For example, a tower having larger upper part and smaller lower part could pose difficulty in localization as what is seen in the input image might be covered or not shown in the 2D map.

Figure 3.2: An Example of Input Image

3.1.2 Prior Information: Two-Dimensional Plan View

As discussed before, the only prior information we have here is a two-dimensional plan view of an urban area. Appearance information is not available. The map is obtained by manually annotating building outlines from a plan view image, such as those from Google Earth [36]. However, interior structural information is not included in the map, such as architectural differences among different floors in a building. In other words, the map only stores the simple exterior contour of a building. For example, differences between high and lower floors in a hotel building due to a swimming pool at middle floor are not captured in the map. The map is stored in the form of 2D locations of the geometric corners.

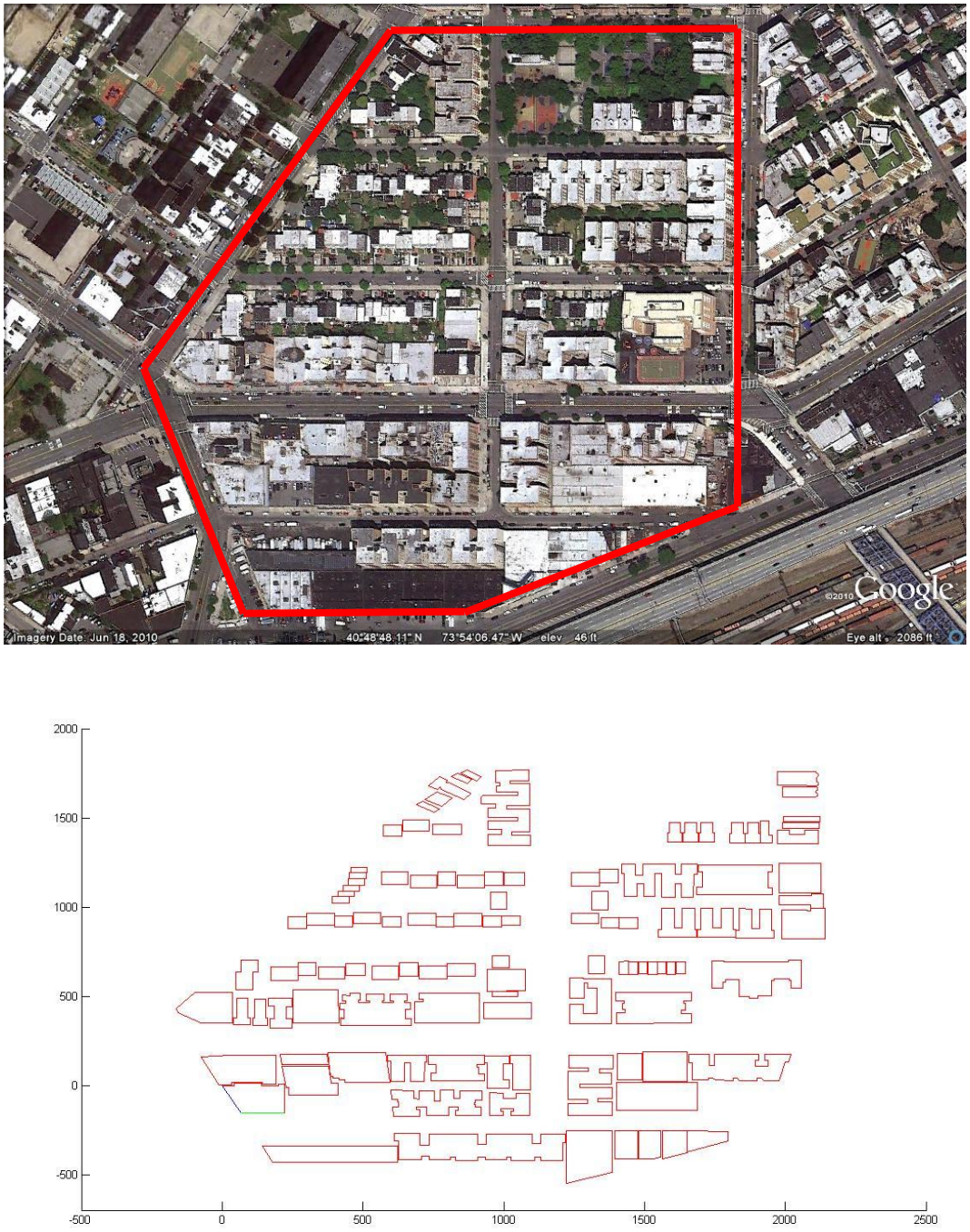


Figure 3.3: Google Earth Plan View and Two-Dimensional Map

3.1.3 Vertical Corner Line Hypotheses (VCLH)

Given an input image of the scene in the 2D map, it is proposed that the localization be done based on building corners. Basically the idea is to establish correspondences between building corners in the image and corner points on the 2D map. A new feature based on building corner is introduced, namely a Vertical Corner Line Hypothesis (VCLH).

A VCLH refers to a hypothesis of a vertical building corner in an image. In particular, it is characterized by position of the vertical line or building corner in the image, and orientations of the neighbouring plane normals. The set of VCLHs extracted from the input image is called a VCLH signature. Matching is performed by identifying the camera position on the 2D map with the closest VCLH signature to the input image one. Figure 3.4 illustrates the concept of matching an image building corner to a map building corner. More details about detection of VCLH and matching are provided in subsequent sections.

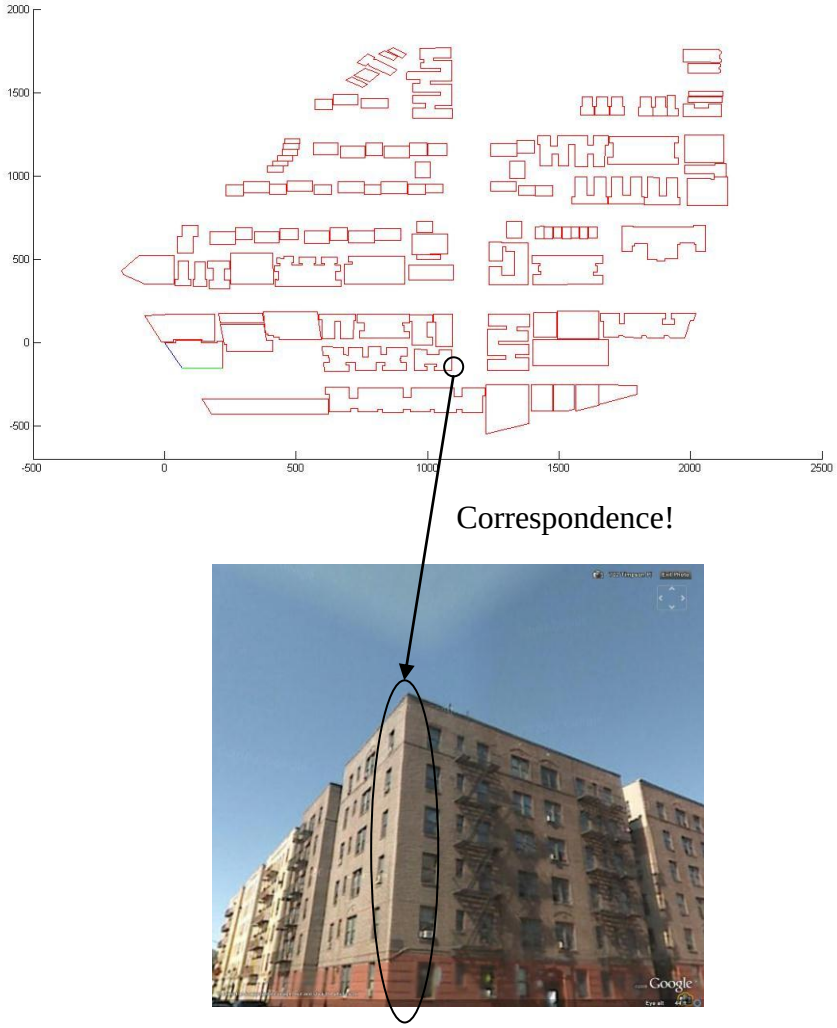


Figure 3.4: Basic Matching Concept

3.1.4 Detection of VCLH

The method for VCLH detection consists of two main stages, detection of Vanishing Point (VP) and strong line segments, and inference of VCLHs. The reason for doing VP detection here is that with information from VP, recovery of three-dimensional (3D) information of line directions becomes possible. Analogous to Marr's two and a half dimensional (2.5D) sketch [37] where local surface orientations are estimated, image lines in this framework also include 3D directions of the line and normals of neighbouring planes.

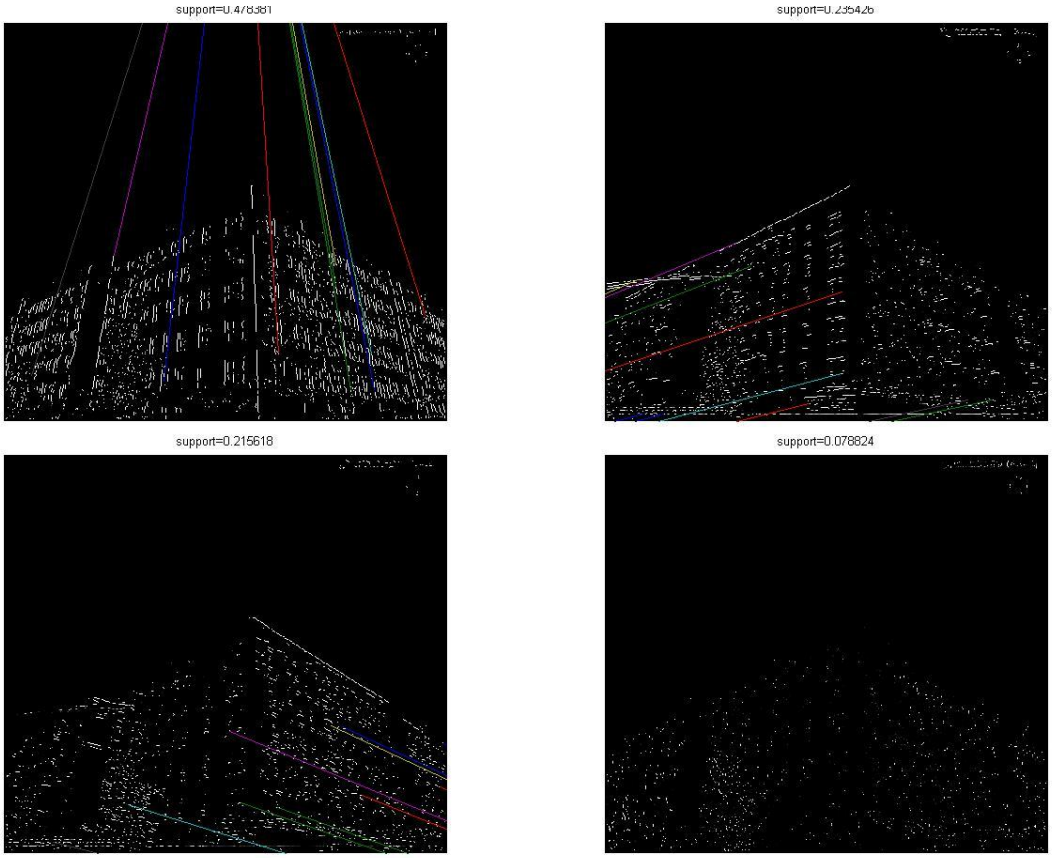
After the VP estimation stage, edges which belong to the same VP and are close together are linked into different line segment. These line segments are used to aid the extraction of VCLH.

3.1.4.1 Detection of Vanishing Point (VP) and Line Segment

Firstly, Canny edge detection with sub-pixel accuracy is performed on the query image. Edges with not enough local linear support are removed. In general, these edges correspond to non-building edges such as those coming from trees and cars. Removal of these leads to more accurate VP estimation.

Positions of VPs are estimated using the Expectation-Maximization (EM) algorithm subsequently, similar to procedures taken in [38] for same purpose. The E-step computes probability of assigning edges to different VPs, while the M-step estimates the locations of the VPs that maximizes the combined log-likelihood.

One important thing to take note here is the initial positions of the Vanishing Points. It has significant effect on the detection results due to the hill-climbing nature of the EM algorithm which is unable to escape from local minima. Repeated random seeding of possible locations of the VPs is used as initial guesses. Figure 3.5 shows an example of the EM process, where intersections of the colour lines are the possible VP locations.



(Top Left and Right, Bottom Left: VP Detection, Bottom Right: Useless Edges)

Figure 3.5: An Example of EM Stage in VP Detection

After VP detection and each edge having been assigned to a VP, line segments are formed from edges by linking edges which are close to each other and removing lines with low edge strengths. Lines which are too short are removed. Sufficiently long lines are used for VCLH detection. Figure 3.6 provides an example of this stage. Different colours are used for edges which belong to different VPs.

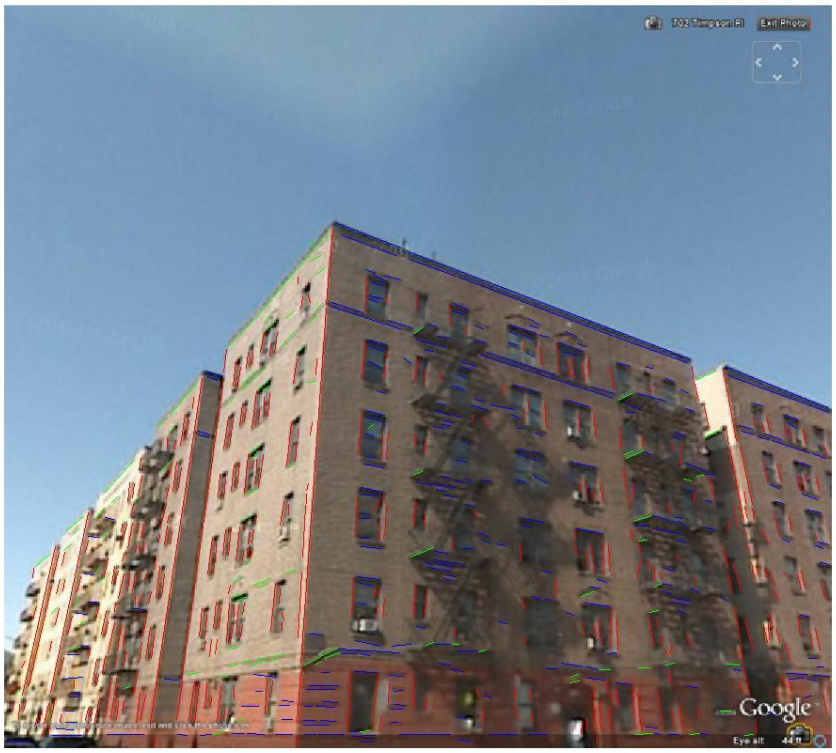


Figure 3.6: Results of Line Segment Detection

3.1.4.2 Rectification

Rectification is carried out next to virtually rotate the camera such that the optical axis is aligned with the ground plane. Vertical lines which are parallel in the scene generally converge to a vertical vanishing point in images. Based on the vertical VP location for which we know the associated 3D direction, a standard rectification homography can be applied to the image to virtually rotate the 3D world normal to be parallel to the camera y-axis. This makes all the vertical lines look parallel to each other. Figure 3.7 shows an example of the rectification process.

Figure 3.7: Rectification Process

3.1.4.3 Estimating Vertical Corner Line Hypotheses (VCLH)

There are three separate conditions for determining a VCLH:

1. Collinearity of vertical line segments,
2. Collinearity of endpoints of horizontal line segments which belong to the same VP, and
3. Collinearity of intersections of extrapolated line segments from different VPs.

(Based on Condition 1 – 3 from Left to Right)

Figure 3.8: VCLH Inference

Next, VCLHs are to be assigned orientations of neighbouring plane normals. For condition 1, there are no orientations. The only support is from a set of collinear vertical line segments. Neighbouring plane normals cannot be determined due to lack of supporting horizontal line segments. For condition 2, the supporting horizontal line segments share the same 3D direction and are also assumed to lie on the same 3D plane. The cross-product between the 3D vertical axis and the 3D unit vector of the horizontal VP direction provides a hypothesis for the 3D normal of the facade plane next to the VCLH. Condition 3 are supported by two sets of horizontal line segments, and a similar analysis to that applied to Condition 2 allows a hypothesis for the normals of the two facade planes on both sides of the VCLH, as shown in Figure 3.7. An example of processing at this stage is provided in Figure 3.8.

Figure 3.9: Results of VCLH detection

As shown in Figure 3.9, a VCLH signature contains positional and orientation information of the VCLHs detected. For each VCLH detected, the one dimensional position (horizontal) and orientations of neighbouring plane normals are stored. Number of orientation planes depends on condition of forming VCLH, as discussed in Section 3.1.4.3. Figure 3.10 below provides an example of VCLH signature stored in the machine.

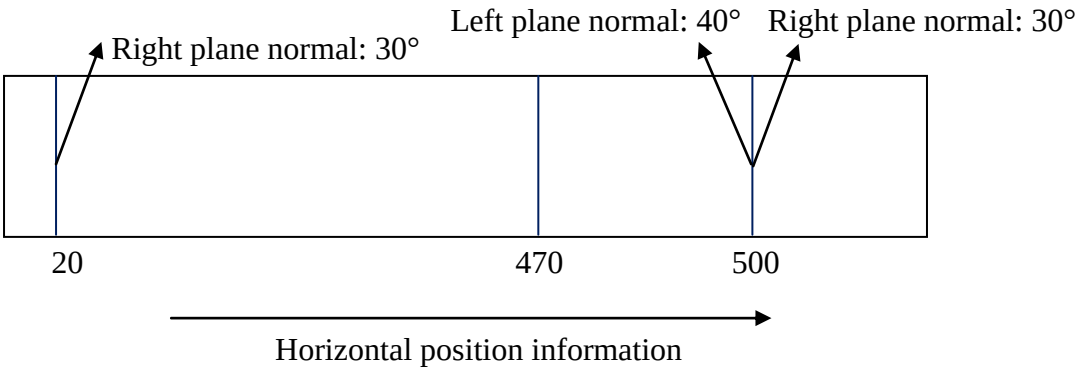


Figure 3.10: An Example of VCLH Signature

3.1.5 Matching: Random Sample Consensus (RANSAC)

In our work, a RANSAC [2] based approach is used whereby search using random sampling is performed and the best camera location and viewing angle is reported. The search is to identify the best correspondence between VCLHs detected in the query image and geometric corners in the 2D map.

Given the 2D map and detected VCLHs from the query image, two VCLHs from the query signature and two geometric corners are randomly selected from the query image and 2D map respectively. The two VCLHs (first hypothesis) selected should be at least certain distance away and have two orientations (neighbouring plane normals). They are used to form a basis for the three Degree-of-Freedom (DOF) transformation comprising of the 2D camera position and a one DOF viewing angle. The basis parameters are positions and plane normals of the VCLHs. The camera location can then be determined through the basis. Camera location here refers to a point (x, y) on the 2D map and viewing angle at that point, total of three DOFs.

Next, an image is created using the two selected geometric corners and the 2D map by standard perspective projection. Differences between plane normals of the VCLHs and geometric corners are used to align the model with viewing angle of the image. The simulated

image is matched against the query and a score is calculated. This process is repeated for a certain number of trials, and the best correspondence is identified. The number of trials must be reasonably large for good confidence level.

It is worth noting here that the use of neighbouring plane normals help reduce the computation cost. Without plane orientation, three VCLHs would be required for establishing the three DOFs described above (three positions are required and hence three VCLHs) and the time complexity of the overall search would be $O(n^3)$. With the use of neighbouring plane normal, the cost is down to $O(n^2)$, saving by an order of magnitude. Furthermore, the other three neighbouring plane normals can be used as an initial test consensus to rapidly eliminate many incorrect bases.

3.2 Speeding up Framework: Geometric Hashing

As discussed before, a RANSAC based search for best camera location is computationally very expensive. For a region with n geometric corners, the number of possible pairs of geometric corners is $O(n^2)$. As the method becomes slow when n is large, Geometric Hashing [3] is employed here to shorten the retrieval time. It is extended to VCLH in this project. Geometric Hashing [3] is selected based on direct similarity of the current problem with the original one defined for the hashing framework. Both problems use geometric information and require tolerance on occlusion and simple geometric transformations.

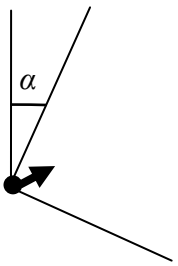
3.2.1 Pre-processing

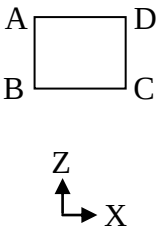
The 2D map is densely sampled at a grid level of 16 metres x 16 metres or 100 x 100 units on the map. Each grid point corresponds to a virtual camera location, which together with an angle of view, forms the simplified camera pose (x, y, α). The other typical camera

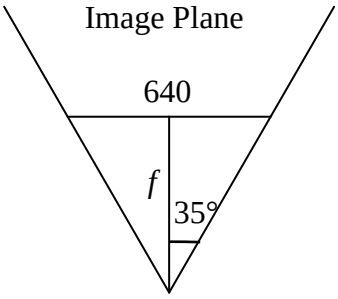
pose parameters have been eliminated through rectification and normalizing for focal length. There are in total 56700 poses. Each camera location is further divided into 63 angles of view, starting from 0 to 6.2 radians with 0.1 radian interval.

For each camera pose, the ideal VCLH signature is generated directly from the map. A one dimensional simulated image is created under perspective projection. VCLH signature is extracted from the image subsequently. Horizontal field of view (FOV) is set at 70° and the image width is set at 640 pixels. Camera locations inside a building and occluded corners from the viewpoint are ignored during the projection process. Plane normals of each VCLH are obtained by subtracting angle of view from the angles the geometric corner makes with the two neighbours.

The VCLH signature obtained is further quantized into 32 levels according to location. For more discriminative power, each bin is further divided into eight bins, this corresponds to quantization of orientations of the VCLHs (angles of neighbouring plane normals, four for left and four for right). The number of VCLH in each bin is counted and combined into a 256 dimensional signature vector. Figure 3.11 provides an illustration of the projection process. Detailed implementation is provided in Algorithm I.







Corner D is not seen with the given camera pose, it is ignored.

Perspective Projection:

$$\tan 35° = \frac{320}{f}$$

$$f = \frac{320}{\tan 35°}$$

$$R = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$$

$$\begin{bmatrix} TX \\ TY \end{bmatrix} = R \begin{bmatrix} X - xc \\ Y - yc \end{bmatrix}$$

$$x = \frac{f}{TY} TX + 320$$

Legend:

θ: angle of view

f: focal length

R: rotation matrix

x: 1D positions of the VCLHs

Plane Normal Calculation:

Corner A is at 0° and C is at 90° of B.

$$LA = 0$$

$$RA = \frac{\pi}{2}$$

$$LN = \frac{\pi}{2} - (LA - \alpha)$$

$$RN = \frac{\pi}{2} - (RA - \alpha)$$

↑ Angle 0 starts at north direction and goes counter clockwise.

Figure 3.11: Creation of Simulated Image

Algorithm I Projection and Extraction of Quantized VCLH Signature

Input: A 2D map with corner point information stored and a camera pose, [*xc*, *yc*, α].

Output: A 256 dimensional quantized VCLH signature, *V*.

Algorithm

Initialize a 256 dimensional vector, *V*, to zeros.

$$f = \frac{320}{\tan 35^\circ} \qquad R = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$$

$$Tx = \frac{640}{32} = 20 \qquad Ta = \frac{\pi}{4}$$

For each unoccluded building geometric corner

$$\begin{bmatrix} TX \\ TY \end{bmatrix} = R \begin{bmatrix} X - xc \\ Y - yc \end{bmatrix}$$

$$x = \frac{f}{TY} TX + 320$$

LA, *RA*: Angles the corner makes with 2 neighbouring corners

$$LN = \frac{\pi}{2} - (LA - \alpha)$$
$$RN = \frac{\pi}{2} - (RA - \alpha)$$

For each VCLH in the image

$$sl = floor\left(\frac{x}{Tx}\right) + floor\left(\frac{LN}{Ta}\right)$$
$$sr = floor\left(\frac{x}{Tx}\right) + floor\left(\frac{RN}{Ta}\right) + 4$$

$$V(sl) = V(sl) + 1$$
$$V(sr) = V(sr) + 1$$

The hash table used here is of two dimensions, with 256 x 50 bins. For visualization, one can view the hash table as a 2D matrix with 256 rows and 50 columns. The first dimension corresponds to each dimension of the VCLH signature. During pre-processing, given an ideal quantized VCLH signature, if there is a VCLH in bin *k*, information of the corresponding camera pose (*x*, *y*, *α)* is inserted into bin k of the 256 bins. This process is repeated for all the 256 bins.

Second dimension of the hash table is used to speed up online retrieval process. It is used to store camera poses with certain number of VCLHs (1-50) in bin k of the 256 bins. For example, given a VCLH signature of a camera pose, if there are two VCLHs in bin 1 of the 256 levels signature, the camera pose information is inserted to bin (1,2). This helps in terms of speed by making online retrieval faster since it avoids the need of repeated access to camera poses with more than one VCLH in the same bin.

This process is repeated for all camera bases. While Geometric Hashing can provide speed gain to the overall camera based localization framework, scalability to a bigger region or map remains an issue as dense sampling over the whole map is required during pre-processing. Algorithm II provides an overview of construction of the hash table.

Algorithm II Construction of Hash Table

Input: A 2D map with corner point information stored.

Output: A two dimensional hash table, H of 256 x 50 bins.

Algorithm

For each camera pose $[x, y, \alpha]$ in the 2D map

If camera pose is inside a building,

Continue

Perform perspective projection and extract quantized VCLH signature, V

For $i \leftarrow 1$ to 256

If $V(i) > 0$

Insert the camera pose $[x, y, \alpha]$ in $H(i)$ $(V(i))$

3.2.2 Online Retrieval

During online retrieval stage, a voting scheme is employed to identify camera poses with closest signatures to the query. If bin k of a given quantized VCLH signature has a value of larger than zero, all camera poses in row k will receive a vote and this voting process is repeated for all 256 bins. Camera poses with scores higher than certain threshold will go through a further verification to identify the best camera pose. Implementation details are provided in Algorithm III.

Algorithm III Online Retrieval

Input

H: Hash Table.

V: A quantized VCLH signature extracted from input image.

T: Threshold for selecting camera poses.

Output

A shortlist of camera poses.

Algorithm

Initialize a 56700 dimensional score vector, *S*, to zeros

For *i* ← 1 to 256

    If *V* (*i*) > 0

        For j ← 1 to 50

            For each camera pose in *H* (*i*) (*j*)

                Increment the corresponding score by 1

Return camera poses with scores larger than *T*

The overall process is equivalent to compare the query signature with all the signatures obtained from the pre-processing stage and select the best. The final score of a particular pose is the accumulation of the scores at each bin, as described below:

$$Score_k = \min(Query_k, Pose_k)$$

$$Final = \sum_{k=1}^{256} Score_k$$

where $Score_k$ refers to score of a camera pose at bin k, $Query_k$ and $Pose_k$ are number of VCLHs at bin k of the query signature of the query and the pre-processed signature of that particular camera pose. *Final* refers to final score (accumulated over 256 bins) of the pose. The voting measure is similar to histogram intersection, whereby the minimum of each corresponding bin in the pair of compared signatures is used.

However, large computation savings can be achieved by this method because of the relatively sparse nature of the query signature since the system only needs to take care of bins with value larger than zero. In general, a VCLH view signature in a city area should be sparse. We would not expect to see so many geometric building corners in a typical urban location. The heavy positional and orientation quantization also help in further sparsity. Experimental results in Chapter 5 have verified this.

In the case of really dense signatures, the voting would be linear to the number of camera poses where access to each bin and camera pose is required. However, pre-processing has reduced the time for online matching as the projection at each camera pose is done offline.

3.2.3 Pyramid

The idea of pyramid or multi resolution is employed to solve the well known problem of bin boundary in the computer vision community. It is based on the idea of weighted Pyramid Matching [39]. Basically the number of bins is extended from 32*8 at smallest spatial bin size when creating the VCLH histogram signature, to include the histogram represented with increasing bin size. This comprises 2^l*8 bins where l is the level number corresponding to a particular spatial bin size. The pyramid extension only applies to the spatial information of the VCLH in the signature, in which the neighbouring plane orientation is simply retained at larger bin size levels.

More specifically, at level l=5, the signature is quantized into 2^5=32 bins as a histogram of the one dimensional locations of the VCLHs. Each bin is further divided into 8 bins for orientation. For level l=4, the VCLH signature is quantized into 2^4=16 bins spatially. Orientation quantization remains the same so there are in total 16*8=128 bins now. This process is repeated until level l=1, where we have only 2^1*8=16 bins. This pyramid framework is carried out at both pre-processing and online retrieval stages. More weights are given to detailed level. The formulation is given below:

$$S(\phi(x),\phi(y)) = \sum_{i=1}^{5} \frac{1}{2^i}\left[I(H_i(x),H_i(y)) - I(H_{i-1}(x),H_{i-1}(y))\right]$$

where ϕ is a histogram pyramid containing signature quantized at different levels, H_i is the level i signature in ϕ, I ($H_i(x)$, $H_i(y)$) is the score at level i, calculated according to the formulation in previous section.

Chapter Summary

This chapter has presented the overall system framework and methodology for the speeding up framework presented. Experimental information and analysis are presented in next chapter.

Chapter 4 Connectivity and Multi Views

This chapter presents details of extension of the existing camera based localization system to the use of connectivity and multi views for further discriminative power. The chapter starts with a brief introduction to a new feature called Structural Fragment (SF) and followed by details of the pre-processing and Geometric Hashing stages.

4.1 Extension of the overall system to Structural Fragment (SF)

As described in Chapter 3, the existing camera based localization system relies on establishing correspondence between input images and the 2D map based on VCLH features. There are few possible extensions from here, including connectivity and omnidirectional view for more discriminative power. Discriminative power here refers to uniqueness level of the visual signature extracted from an image or a location. In general, higher discriminative power or uniqueness level would increase the performance of the system as there is more information for localization now.

4.1.1 Structural Fragment (SF)

A new feature called Structural Fragment (SF) is introduced by grouping VCLHs that are connected. Under certain conditions, it can be established that two VCLHs are connected by a planar facade. This facade would correspond to a straight portion of a building outline in the 2D map. The additional complexity of the SF feature leads to greater discriminative power, and also imposes a topological ordering of the correspondences.

4.1.2 Multi View

To further extend the discriminative power, we incorporate multiple views into the system. Four images are taken at the same camera viewpoint but at different viewing angles to capture more information for more discriminative power. Figure 4.1 below provides an example of use of SF and multi view.

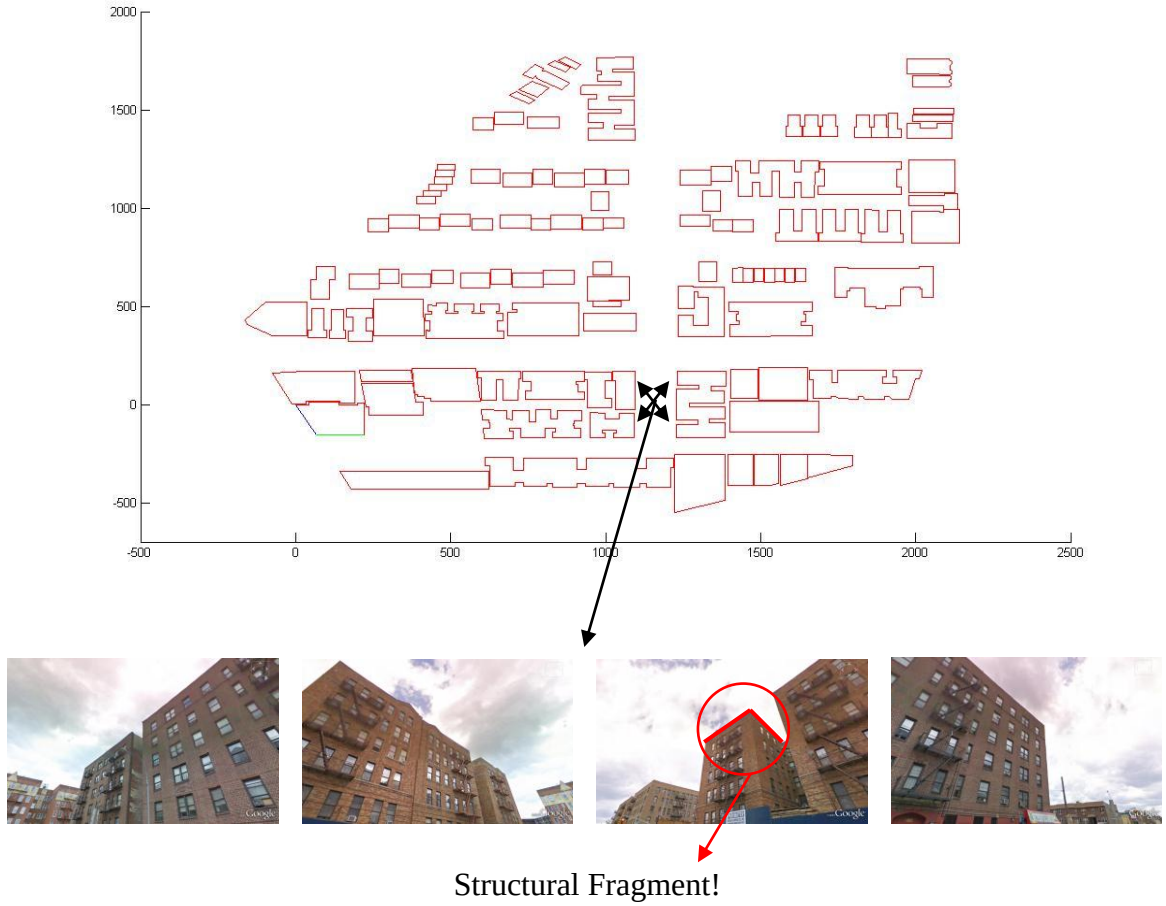


Figure 4.1: Extension to Structural Fragment and Multi-View

4.1.3 Extraction of Structural Fragment (SF) Feature

After a VCLH signature is extracted from the input image, two more steps are taken to establish SFs. Pairs of VCLHs that share common neighbouring plane normals and are

linked by horizontal segments are identified – these VCLH pairs form the elemental planes. Furthermore, elemental planes are linked in a subsequent stage. Figure 4.3 provides two examples of SF extraction.

4.1.3.1 Determining Elemental Planes

VCLHs that are formed based on collinearity of endpoints of horizontal line segments which belong to the same VP, or collinearity of intersections of lines from different VPs are further investigated for connectivity. If two VCLHs share a big enough common set of supporting horizontal line segments, the two VCLHs are considered connected and an elemental plane is formed.

4.1.3.2 Invariant In-Plane Depth Ratio

There is an underlying property that we can exploit to accurately estimate the exact shape of an SF. Given two connected VCLHs on an elemental plane L, the ratio of their depths $\frac{z_a}{z_b}$ is invariant to the perpendicular distance of the line from optical centre and may be computed directly from the image so long as the 3D normal of the elemental plane is known. The detailed mathematical formulation is provided below.

Based on the diagram in Figure 4.2, we consider a VCLH that is observed in the image and can be positioned by the angle ϕ, assuming that the camera focal length has been found through the earlier camera calibration process. The depth of this VCLH, z, is unknown. Additionally, the VCLH lies on the elemental plane L for which the 3D orientation is known, and in the rectified 2D framework the orientation can be described as the angle θ. For two VCLHs that lie on a common elemental plane, the ratio of their depths is invariant to the actual depths of the VCLHs:

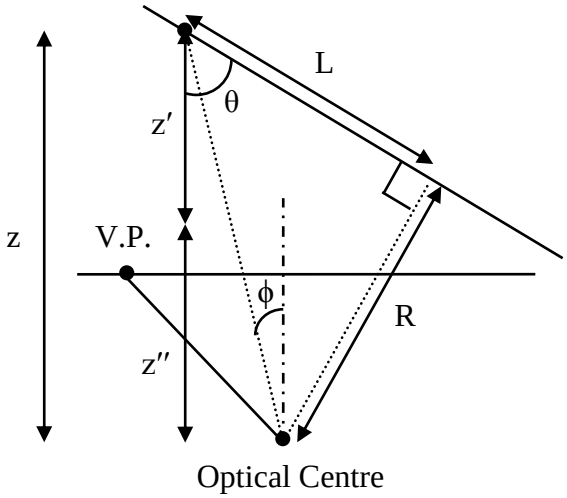


Figure 4.2: Illustration of Invariant In-Plane Depth Ratio

Invariant Depth Ratio:

$$\frac{z_a}{z_b} = \frac{\tan(\frac{\pi}{2} - \theta + \phi_a)\cos\theta + \sin\theta}{\tan(\frac{\pi}{2} - \theta + \phi_b)\cos\theta + \sin\theta}$$

Proof:

$$z' = L\cos\theta$$
$$z'' = R\sin\theta$$
$$\Rightarrow L = R\tan(\tfrac{\pi}{2} - \theta + \phi)$$
$$z = z' + z''$$
$$\Rightarrow z = R[\tan(\tfrac{\pi}{2} - \theta + \phi)\cos\theta + \sin\theta]$$
$$\therefore \frac{z_a}{z_b} = \frac{\tan(\frac{\pi}{2} - \theta + \phi_a)\cos\theta + \sin\theta}{\tan(\frac{\pi}{2} - \theta + \phi_a)\cos\theta + \sin\theta}$$

4.1.3.3 Constructing SF

A further step is taken to link elemental planes with shared VCLHs to form an SF. It is important to note here that the depth ratios for VCLHs on differently angled planes are

different. The unknown of a reference VCLH depth is taken as a common baseline for the relative depths to enable to linking process. The formula used is:

$$z_k = z_0 \prod_{i=1}^{k} \frac{\tan(\frac{\pi}{2} - \theta + \phi_i)\cos\theta + \sin\theta}{\tan(\frac{\pi}{2} - \theta + \phi_{i-1})\cos\theta + \sin\theta}$$

Figure 4.3: Examples of SF Estimation

4.1.4 Localization

After SF extraction, for each SF, a linear search over all possible candidates in the 2D map is carried out. One thing to take note here is that there is a need for a way to combine the best matches from different SFs and report the final best camera location for the query image. A voting-based camera pose estimation algorithm is developed to address this [1].

4.2 Extension of Geometric Hashing to Structural Fragment (SF)

The new Geometric Hashing idea is similar to the original one [3]. However, the hash keys used here are SFs. During the offline pre-processing stage, given a building model extracted from the 2D map with *n* linked points in 2D Cartesian Space, two points are taken as basis and a coordinate system is formed. Coordinates of the remaining *(n-2)* points in this new coordinate system are determined subsequently. Next, information of the model and

basis pair is inserted into the corresponding bins of a 2D hash table. A model here refers to a series of connected building corners or points in the 2D map. This process is repeated for all models and possible basis. Figure 4.4 provides an overview of pre-processing stage.

For example, given an SF with five linked points, two points are taken as basis to establish a coordinate system. Coordinates of the remaining three points are determined using this coordinate system. Assuming there is a point at (u, v), the model and basis information are inserted into bin (u, v) of the 2D hash table. The model and basis information typically looks like corner: 10 to 14, basis: corner point 10 and 11. Same procedures are taken for the remaining two points and the overall process is repeated for all possible pairs of basis from the five points, which is ${}^{5}C_{2}$=10 combinations.

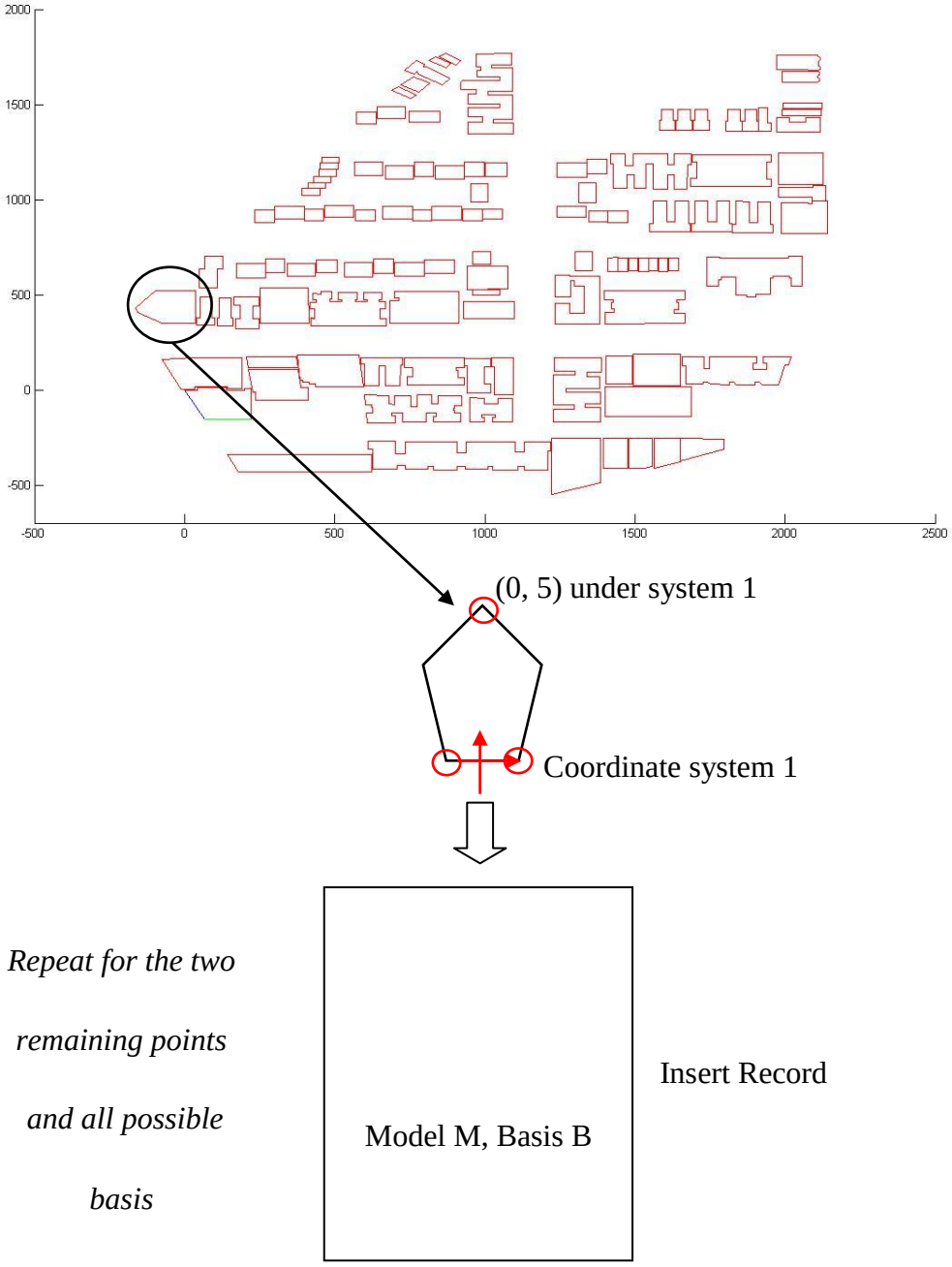


Figure 4.4: Pre-processing Stage of Geometric Hashing

During online retrieval, given a detected SF, similar procedures are taken. Two points are taken as basis to define a coordinate system and coordinates of the remaining points are

determined accordingly. Next, a voting based mechanism takes place to identify the model basis pair with votes over a threshold. These model basis pairs are passed to the verification stage for further analysis. In the current implementation, the threshold is set to maximum which means for an n points SF, the threshold is set to n - 2. In general, a higher threshold is more selective and hence more candidates are discarded during Geometric Hashing stage.

Alternatively, the hashing process can be viewed as a shortlisting mechanism. For each point besides the basis, the system returns a shortlist of good possible candidates. Hence, for an SF with n points, setting the threshold to n - 2 is equivalent to taking intersection of the $(n - 2)$ shortlists returned by the $(n - 2)$ points excluding basis since a candidate is dropped if it does not exist in any one of the shortlist.

For the same example as above, given the SF with five linked points, two points are taken as basis to establish a coordinate system. Coordinates of the remaining three points are determined using this coordinate system. Assuming there is a point at (u, v), all model basis pairs information in the bin (u, v) of the 2D hash table are extracted out and each of them gets a vote. Same procedures are taken for the remaining two points.

A direct implementation would take $O(n^2)$ steps to complete pre-processing for a building with n corners. In fact, topological order of SF can provide further speedup by an additional restriction of establishing basis only from two consecutive points, both during pre-processing and online retrieval stage. It is obvious that if we restrict ourselves to taking two consecutive points to establish basis during online query, pre-processing stage only needs to take care of bases that are formed by two consecutive points. For a building with n corners, the number of bases to be processed is n. The overall cost is brought down to $O(n)$.

Incorporation of SF into the Geometric Hashing framework avoids the need of dense sampling over the whole map and hence greatly increases the pre-processing and retrieval speed due to much smaller number of bases, from 56700 (number of camera poses) to 885

(number of geometric corners on the map). Algorithmic details are provided in Algorithm IV and V.

Algorithm IV Construction of Hash Table using SF

Input: A two dimensional map with corner point information stored.

Output: A two dimensional hash table, *H*.

Algorithm

For each building in the map

Repeat

Select two corners of the building as basis.

Establish a new coordinate system using the two points chosen.

For each remaining points in the building

Calculate the corresponding coordinate (*u*, *v*) under the new system.

Insert record (building index and basis index) into bin (*u*, *v*) of the hash table.

Algorithm V Online Retrieval using SF

Input

H: Hash Table.

SF: An SF signature extracted from input image.

T: Threshold for selecting candidates for further verification.

Output

A shortlist of SF candidates.

Algorithm

Initialize a score vector for all possible SF candidates to zeros.

Select two corners of the building as basis.

Establish a new coordinate system using the two points chosen.

For each remaining points in the building

Calculate the corresponding coordinate (u, v) under the new system.

For each record in bin (u, v) of H

Increment the corresponding score by 1.

Return candidates with scores larger than T

Chapter Summary

This chapter has presented the extension of the system framework and methodology to the idea of connectivity and omnidirectional view. A new feature called Structural Fragment (SF) has been introduced. Experimental information and analysis are presented in next chapter.

Chapter 5 Experiments and Analysis

Now that the speeding up framework has been established, this chapter will investigate the performance of the framework. Experimental setup is discussed at first, and a study of theoretical upper bound on achievable performance (uniqueness test) using VCLH and SF follows. Lastly, detailed experimental results and analysis are presented. Experimental results are divided into two parts, VCLH based and SF based.

5.1 Experimental Setup

The prior information available to the system is a 2D plan view of an urban area. An area of 440m by 440m metres in New York (Bronx Area) (40°48'46"N, 73°54'5"W) is selected for testing of the system. The map is prepared by manually annotating plan views obtained from Google Earth [36], as shown in Figure 5.1.

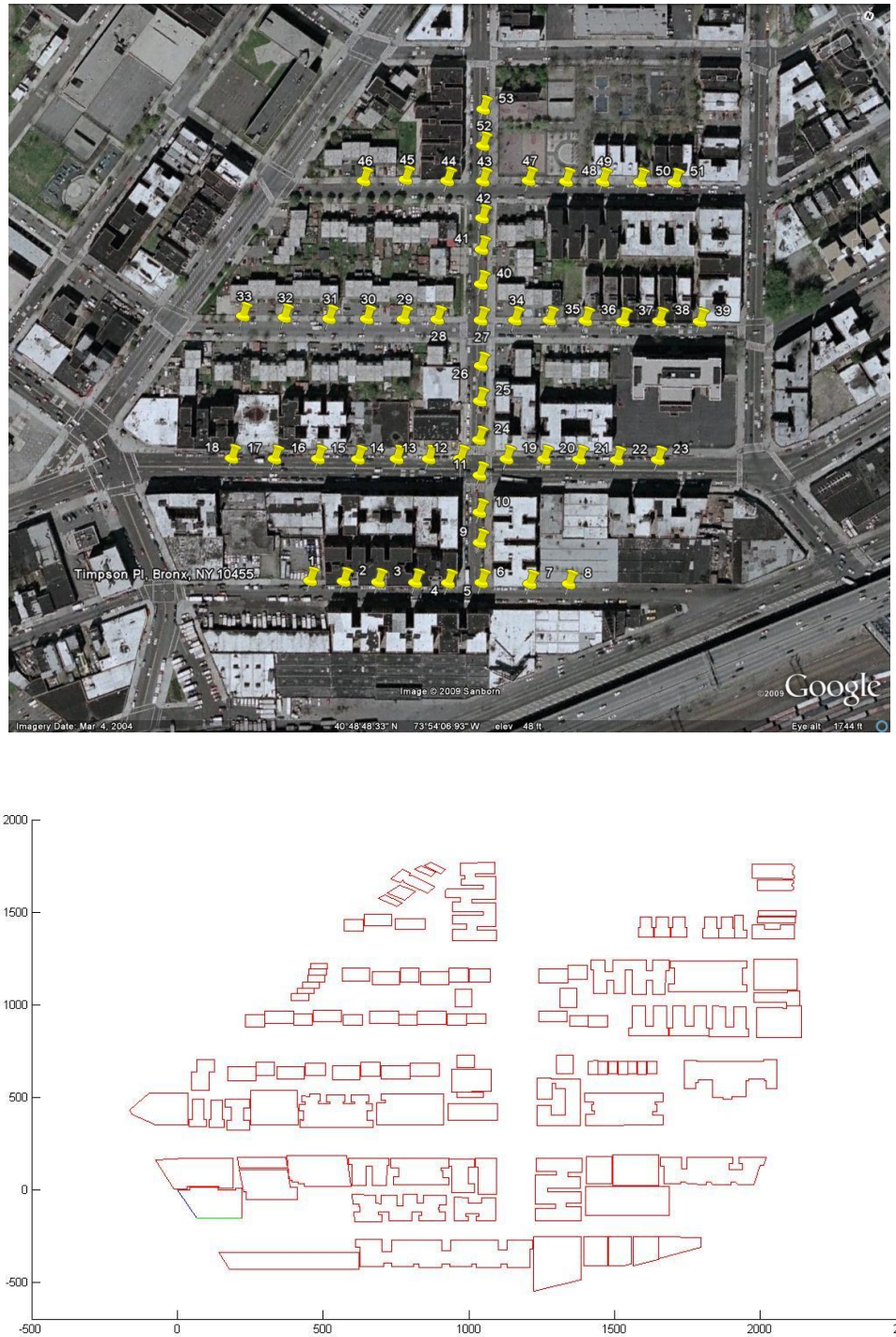


Figure 5.1: Google Earth Plan View and Two-Dimensional Map

A dataset containing 212 images was prepared from Google Earth [36] within the region, with camera location recorded. 53 locations were uniformly sampled from the map and four images were extracted from each location, at viewing angle of 45° from the road for each quadrant. 45° was selected to capture sufficient view information. The 2D plan view in Figure 5.1 and Figure 5.2 below shows sampled location and dataset in a collage form (yellow pin refers to a camera location and images in the same black box were taken from the same location but different viewing angles). The images were taken as input to the system and the output of the Geometric Hashing system was scanned through to see if the ground truth was within the list.

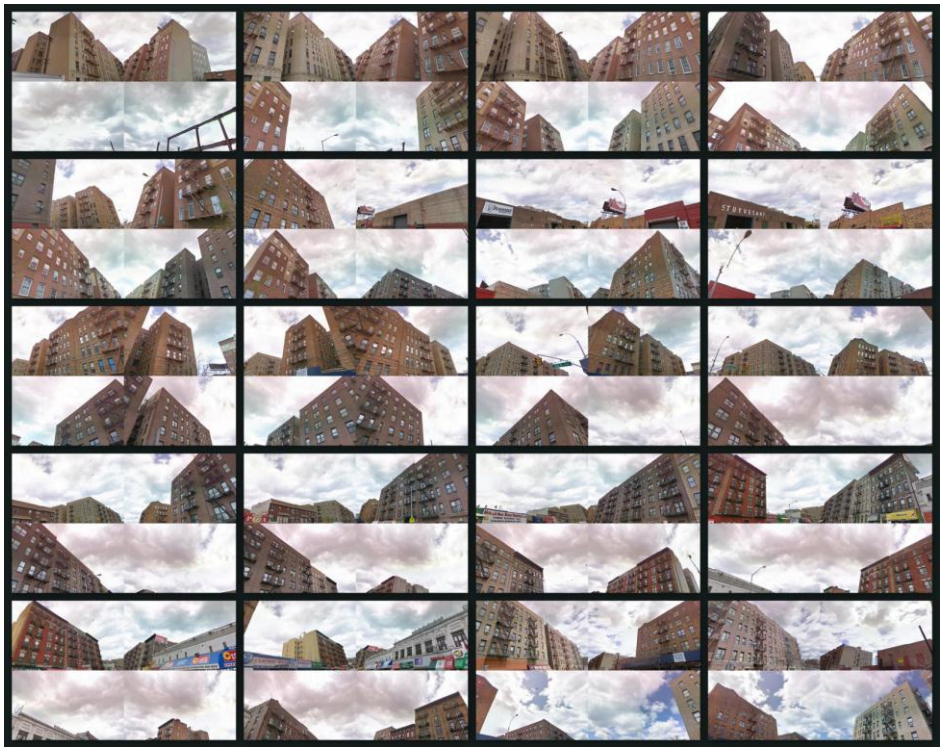


Figure 5.2: Location Sampling and Test Dataset

The 2D map was densely quantized into 900 bins at the resolution of 100 x 100 in map scale, or 16m x 16m in real world scale. Scoring was done exactly as discussed in previous chapters. Implementation was done in Matlab [4].

5.2 Uniqueness Test of VCLH and SF Signatures

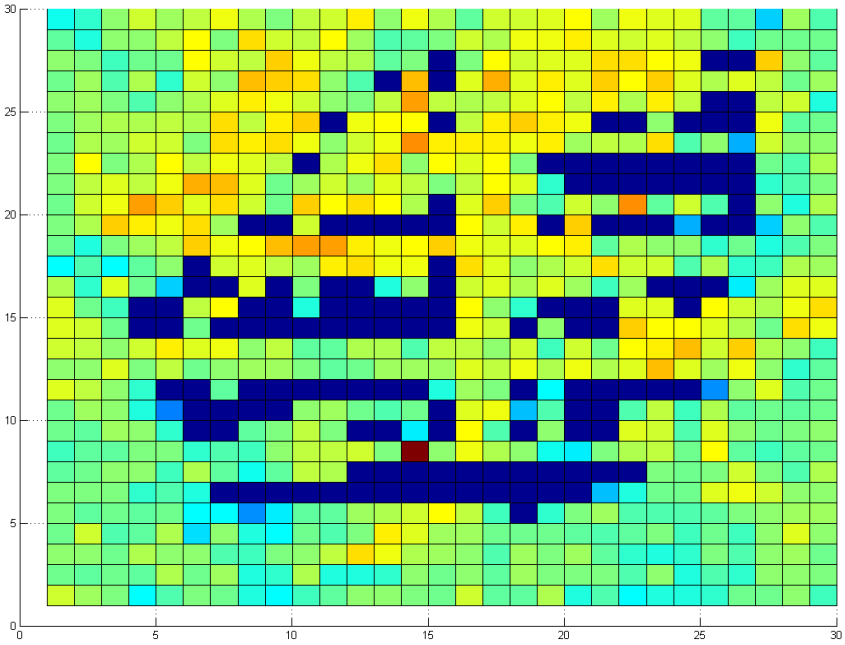
A uniqueness test was carried out to investigate how distinctive the signature of a basis is from the other signatures, assuming signature was extracted correctly from the image. The experimental results reveal that despite human perception that different parts of a bland urban area look the same, the geometric VCLH and SF signatures are surprisingly unique.

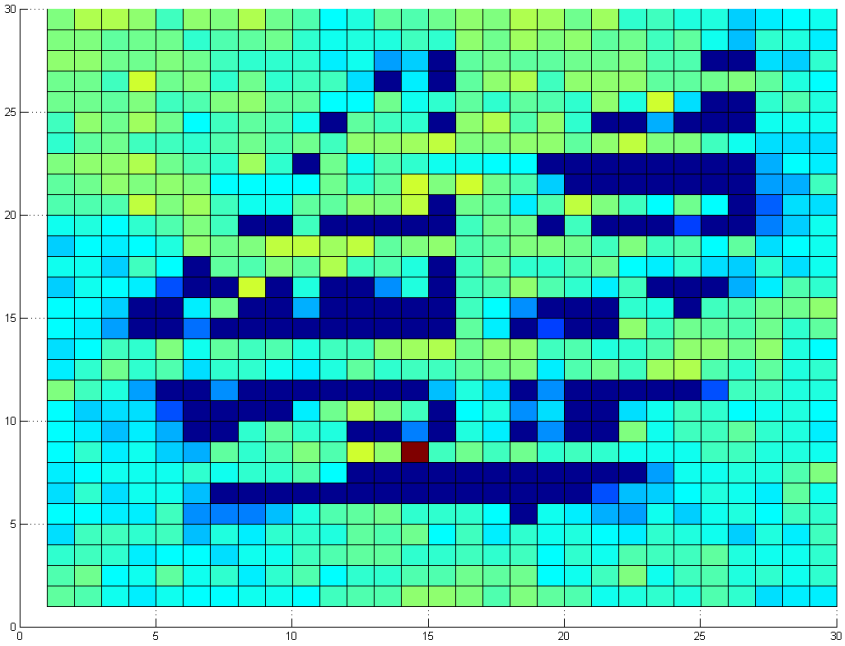
The test created a VCLH signature directly from the 2D map at an arbitrarily specified location. The signature was used as input to the Geometric Hashing system subsequently and the scores of each basis against the query signature were recorded and compared. The more different the ground truth is from the rest, the better it is.

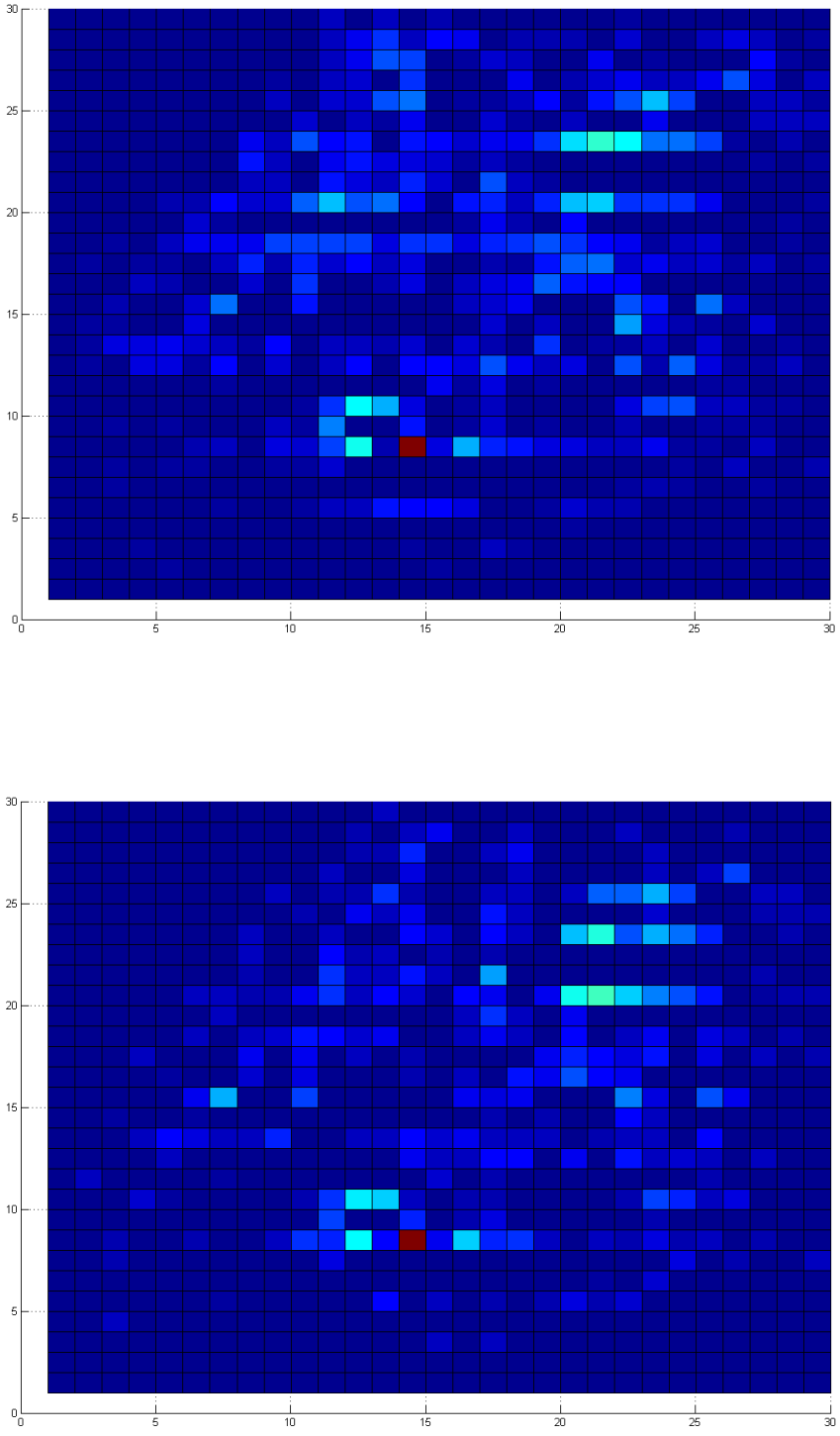
Figure 5.3 shows some examples of the signatures and how different they are from the rest. These are the score maps given a query signature. Red colour indicates high score for that particular basis and blue indicates low score. In other words, the more bluish the score map is, the better it is. As discussed before, the ground truth is guaranteed to have the maximum score in the map as the distance measure is based on Histogram Intersection.

Next, the test was also extended to omnidirectional images. Basically, few images taken at the same location but with different viewing angle were used as query, each image was scored independently and all the scores were summed. Omnidirectional images improve the system performance because more corners can be seen in most of the cases and the positions carry much more discriminative power or uniqueness. The results are same as what is expected, with the score maps look much more bluish with omnidirectional inputs. This is desired as the signatures are really distinctive.

Lastly, investigations on use of elemental planes and SF were also carried out. The results show that both elemental planes and SF bring further discriminative power.


0
5
10
15
20
25
30


0
5
10
15
20
25
30



(From Top: Lines, VCLH, Elemental Plane and SF)

Figure 5.3: Results of Uniqueness Test

5.3 Experimental Results - VCLH Based

After the image pre-processing stage, VCLH signatures were provided to the Geometric Hashing system and voting based camera pose estimation to search for best camera location. The experiments looked at position of the ground truth (camera location) in a shortlist returned by the final detailed verification stage containing top 30 camera locations for a query image (3.33% selectivity, 30 out of 900). Scores from the same camera location but different viewing angles were summed together. The experiments were carried out on all images in the dataset presented above.

The generation of the 1D geometric hash table took approximately fifteen minutes to complete, for 100 x 100 resolution in map scale. A typical online retrieval completed within ten seconds, with selectivity of 11.11% (length of shortlist is 100 out of 900). The speed gain of Geometric Hashing is significant. A direct search using RANSAC could take more than ten minutes to complete. Geometric Hashing was able to remove a large pool of candidates fast. However, the pre-processing time depends heavily on grid resolution. A 100 x 100 in map scale sampling requires one to two hours to complete. Scalability to a bigger map is possible, but computation cost is still high.

In the experiments, none of the ground truths made it to top 30 rank, with or without Geometric Hashing. The results are not satisfying. It was observed that the performance of the VCLH detection stage was quite bad and the number of false positives was very high, this caused the query signature to be very far from the ground truth. Figure 5.4 below provides two examples of output of the VCLH detection stage. Two examples of score maps obtained based on actual VCLH signatures extracted from query images are shown in Figure 5.5. Each square maps to a camera location and black dot refers to ground truth. At each camera location, we compute the score between the VCLH signature obtained at camera location and the signature at the ground truth location. Square which are reddish have higher scores indicating close match, while squares which are bluer have lower scores. The score maps provide an indication as to how discriminative the ground truth is with the preferred situation

being a red square at the ground truth location with all other squares being blue. As the ground truths did not make it to top rank in these examples, this is undesirable.

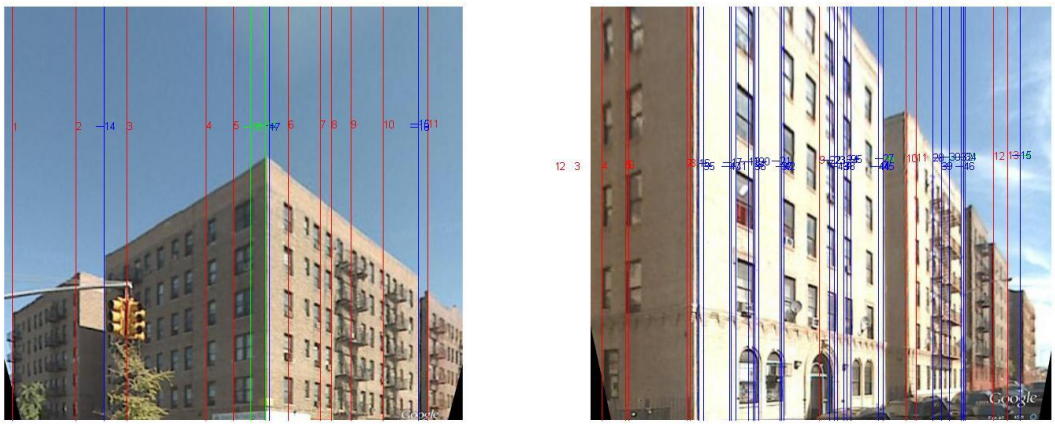


Figure 5.4: Detections of VCLH

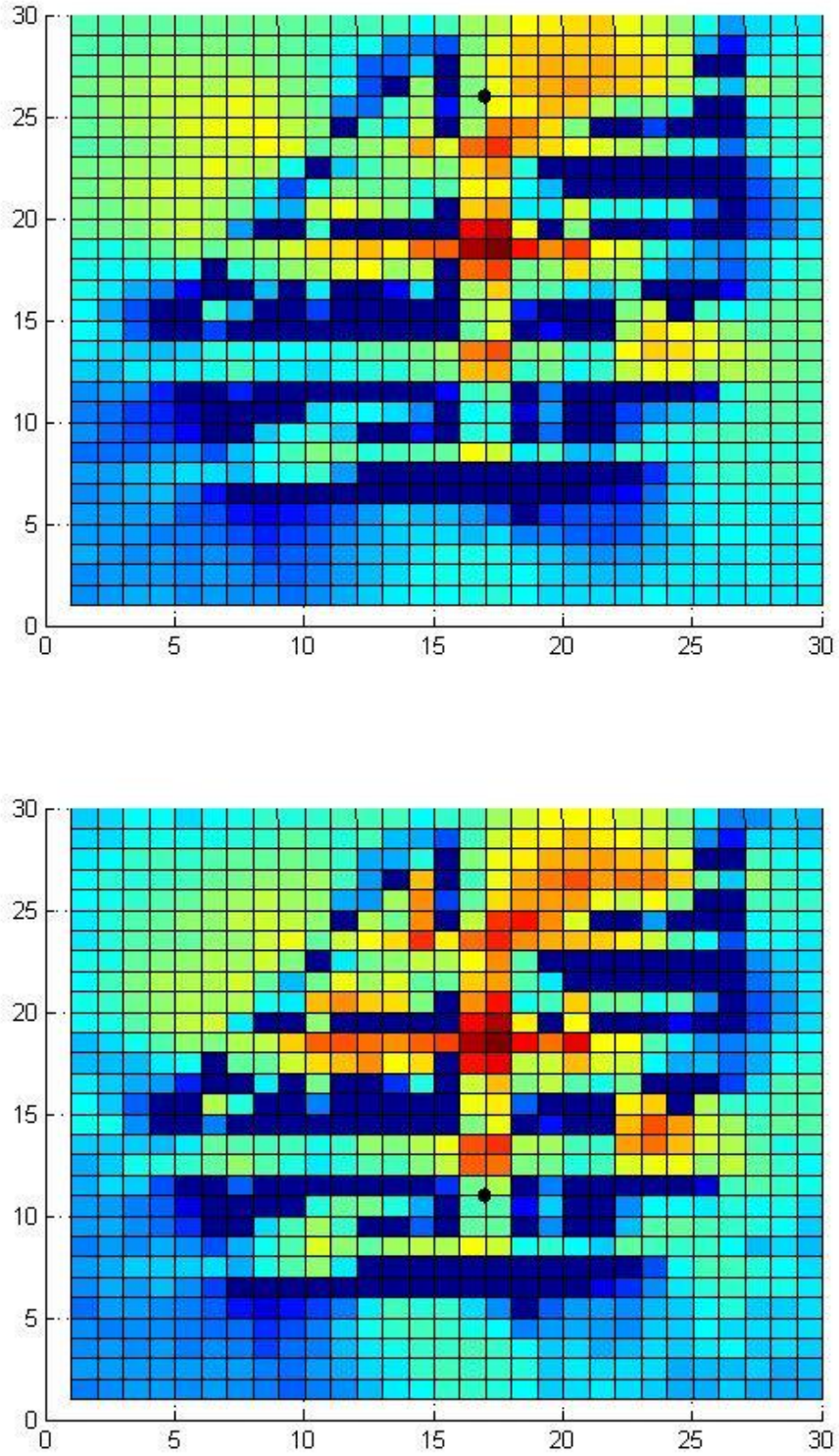


Figure 5.5: Score Maps of VCLH Based Geometric Hashing

5.4 Experimental Results - SF Based

After the image pre-processing stage, SF signatures (SFs detected in query images of the dataset) were provided to the Geometric Hashing system and voting based camera pose estimation to search for best camera location. The experiments looked at position of the ground truth (camera location) in a shortlist returned by the voting based camera pose estimation containing top 30 camera locations for a query image (3.33% selectivity, 30 out of 900). Scores from the same camera location but different viewing angles were summed together. The experiments were carried out on all images in the dataset presented in section 5.1.

The pre-processing (Geometric Hashing, generation of 2D hash table) Matlab code took about 30 seconds to complete. The speed gain compared to the VCLH based system is significant, because there is no need for dense sampling of the 2D map during pre-processing and hence the scalability to larger region or map is much better. A total of 885 bases were processed instead of 56700 bases in the VCLH based system. An online retrieval typically takes less than a second. Figure 5.6 provides an example of the Geometric Hashing online retrieval stage.

A typical direct matching (voting based camera pose estimation) without Geometric Hashing took eight seconds to complete and four seconds with matching. In our unoptimized Matlab code, the speed gain is significant.

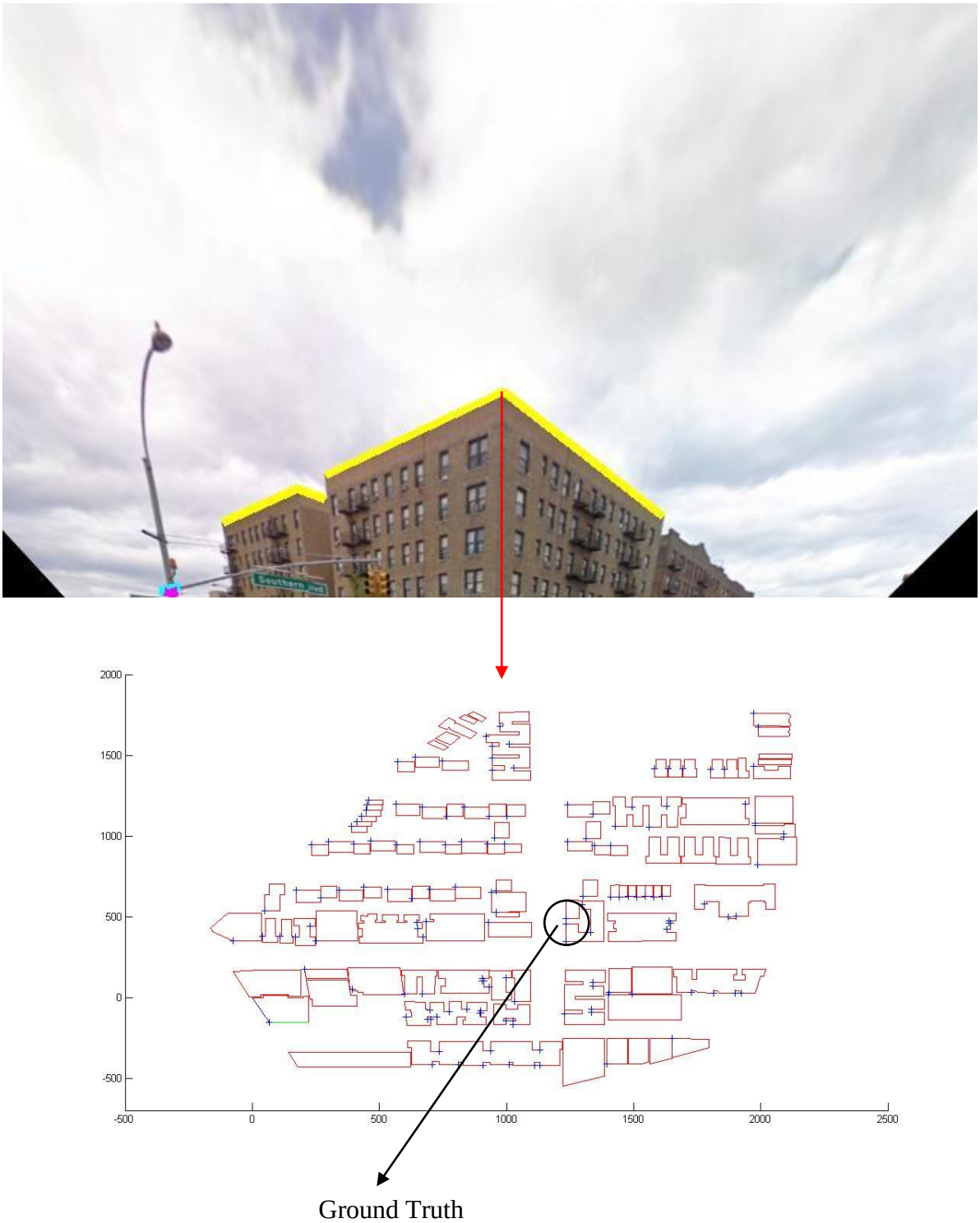


Figure 5.6: An Example of Online Query Stage

With Geometric Hashing, 41.51% of the cases have top 30 shortlists of ground truth camera locations. Detailed experimental data is presented in Table 5.1. The table compares performance of Geometric Hashing with direct matching (without Geometric Hashing) in terms of number of correct matches falling into top 30 out of 900 bins. The top 30 ranks are divided into 7 categories and number of correct matches falling into each is shown. Cumulative (%) refers to cumulative percentage of correct matches at each category such as percentage of correct matches falling into top 15.

Rank	1	2-5	6-10	11-15	16-20	21-25	26-30
With Geometric Hashing	0	3	2	3	6	5	3
Cumulative (%)	0	5.66	9.43	15.09	26.42	35.85	41.51
Without Geometric Hashing	2	9	3	4	4	2	3
Cumulative (%)	3.77	20.75	26.42	33.96	41.51	45.28	50.94

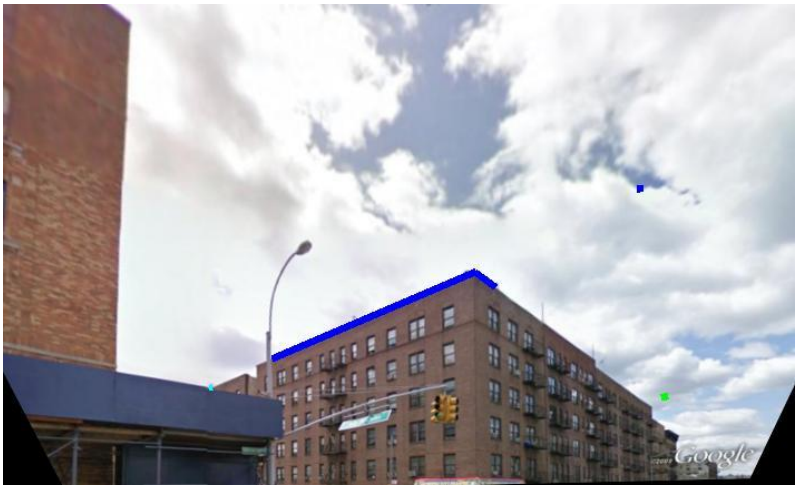
Table 5.1: Performance of Geometric Hashing and Direct Matching

Next, study on selectivity of Geometric Hashing was carried out. The selectivity is in the range of 10 % to 30 % out of 885 possible candidates in the 2D map. Selectivity here refers to the ratio of number of candidates selected by geometric hashing stage to the total number of candidates given an SF as input. Ideally the selectivity should be as low as possible as we do not want too many candidates to pass the stage. Theoretically, longer SF carries more information and hence the selectivity is lower. Table 5.2 provides some experimental information on selectivity of the framework.

Number of corners in SF	3	4	5
Length of shortlist	212.98	91.06	113.33
Selectivity	24.1%	10.3%	12.8%

Table 5.2: Study on Selectivity

The results are quite satisfying. However, there are still rooms for improvements. As shown above, Geometric Hashing brings down the overall performance by a little. This is because some correct matches have been filtered out during the hashing stage. There are two main reasons for this, inaccurate SF estimation and view occlusion. Figure 5.7 provides two examples of wrong SF estimation. The first is an example of inaccurate SF estimation. The second shows both cases of inaccurate SF estimation and the occluded case.



(a)

(b)

Figure 5.7: Examples of Wrong SF Estimation

In the current system, the threshold is set at (n - 2) for an n - linked SF. A possible solution is to lower the threshold or increase bin size of the hash table. However, this is done at the cost of more false positives. Worst case retrieval time would be linear to number of candidates, which is still acceptable in this case. Another possible solution to view occlusion would be a position and direction based hashing. Although the positional information of the end points might not carry much information desired, the direction could be used to achieve

more discriminative power. For example, in a 2D coordinate system, it is known that the end points of the SF are on certain directions from the origin. Hence bins that lie along those directions get a vote or weighted vote of less than one indicating weaker evidence. Ultimately, performance of the system is highly dependent on successful and accurate detection of SF.

Lastly, although Geometric Hashing brings down the overall system performance by some amount, we believe that the problem of non-zero false negatives can be addressed as analyzed above. It is of significant interest to improve the search to sublinear time complexity as it improves the scalability and certainly, real time performance.

Chapter Summary

This chapter has presented the experimental information and analysis of the Geometric Hashing systems developed. Next chapter is a conclusion to this master thesis with discussion of possible directions for future works.

Chapter 6 Conclusion and Future Works

A Geometric Hashing framework has been established to speed up the current system and the final system gives good performance and runs fast. Experiments and analysis show that there is also an upper bound on the performance of geometric hashing based on VCLH alone and the pre-processing stage is computationally expensive. However, the problems are solved by extending the features used to SF.

Being different from appearance based approaches such as SIFT [40], the framework only relies on simple geometric information and does not require any apriori appearance information on the 2D map. 3D information is not required as well. This improves the usability of the localization framework as appearance and 3D information are not always available such as in military missions in enemy territory.

On the other hand, there are certain limitations with the framework. The core idea of the framework is to establish a correspondence between geometric visual signatures obtained from the query image and 2D map. However, architecture of the buildings has significant influence on the performance. Irregular architectural designs such as tower and sphere – like shape would pose big challenge if details cannot be seen from the 2D plan view. Certainly, a successful localization also requires enough geometric visual signatures to be extracted from the query image. In other words, it relies on successful pre-processing stages of query image such as vanishing point detection and feature extraction.

There are few possible directions to extend the current system and this project. Investigation into more discriminative features to be used is definitely of significant importance here. It would also be interesting to look at some other existing speeding up frameworks such as those mentioned in Chapter 2 Literature Review, including Locality

Sensitive Hashing (LSH) or Parameter Sensitive Hashing (PSH). Further performance gain might be possible by using these state-of-the-art methods.

Although the detection results with SF signatures are a lot better than VCLH signatures, the errors are still quite significant. There is a need to look into the building detection part of the system. Possible features for learning include SIFT [40] or rotatable Haar [41] to cater for perspective distortion.

It is observed that errors of Vanishing Point (VP) detection using state of the art methods are still significant due to noises coming from non-building components such as cars, lamp posts and pedestrians. There is a need for improving VP detection. One point worth noting here is that most VP detection methods require certain assumptions on number of VPs or orthogonality. Hence, developing a general yet accurate VP detection method is of significant interest.

References

1. Tat-Jen Cham, Arridhana Ciptadi, Wei-Chian Tan, Minh-Tri Pham, Liang-Tien Chia. *Estimating Camera Pose from a Single Urban Ground-View Omnidirectional Image and a 2D Building Outline Map*. in *Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition*. 2010. San Francisco.
2. Fischler, Martin A. and Robert C. Bolles, *Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography*, in *Readings in computer vision: issues, problems, principles, and paradigms*1987, Morgan Kaufmann Publishers Inc. p. 726-740.
3. Lamdan, Y. and H. J. Wolfson. *Geometric Hashing: A General And Efficient Model-based Recognition Scheme*. in *Computer Vision., Second International Conference on*. 1988.
4. *Matlab*. Available from: http://www.mathworks.com/products/matlab/.
5. Indyk, P. and R. Motwani. *Approximate nearest neighbors: towards removing the curse of dimensionality*. in *Proceedings of STOC98: 13th Annual ACM Symposium on Theory of Computing, 23-26 May 1998*. 1998. New York, NY, USA: ACM.
6. Lepetit, V., P. Lagger, and P. Fua. *Randomized trees for real-time keypoint recognition*. in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. 2005.
7. Moosmann, F., E. Nowak, and F. Jurie, *Randomized clustering forests for image classification.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008. **30**(Copyright 2008, The Institution of Engineering and Technology): p. 1632-46.
8. Tsai, Frank C. D., *Geometric hashing with line features.* Pattern Recognition, 1994. **27**(3): p. 377-389.
9. Gavrila, D. M. and F. C. A. Groen, *3D object recognition from 2D images using geometric hashing.* Pattern Recognition Letters, 1992. **13**(Copyright 1992, IEE): p. 263-78.
10. Tzong-Chen, Wu and Chang Chin-Chen, *Application of geometric hashing to iconic database retrieval.* Pattern Recognition Letters, 1994. **15**(Copyright 1994, IEE): p. 871-6.
11. Gueziec, A. and N. Ayache. *New developments on geometric hashing for curve matching*. in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 15-17 June 1993*. 1993. Los Alamitos, CA, USA: IEEE Comput. Soc. Press.
12. van Dijck, H. and F. van der Heijden, *Object recognition with stereo vision and geometric hashing.* Pattern Recognition Letters, 2003. **24**(Copyright 2003, IEE): p. 137-46.
13. Mokhtarian, F. and A. K. Mackworth, *A theory of multiscale, curvature-based shape representation for planar curves.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992. **14**(Copyright 1992, IEE): p. 789-805.
14. Mowltarian, F., N. Khalili, and P. Yuen, *Multi-scale free-form 3D object recognition using 3D models.* Image and Vision Computing, 2001. **19**(Copyright 2001, IEE): p. 271-81.

15. Johnson, A. E. and M. Hebert. *Recognizing objects by matching oriented points*. in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 17-19 June 1997*. 1997. Los Alamitos, CA, USA: IEEE Comput. Soc.
16. Bebis, G., M. Georgiopoulos, and N. V. Lobo, *Using self-organizing maps to learn geometric hash functions for model-based object recognition.* IEEE Transactions on Neural Networks, 1998. **9**(Copyright 1998, IEE): p. 560-70.
17. Rigoutsos, I. and R. Hummel, *A Bayesian approach to model matching with geometric hashing.* Computer Vision and Image Understanding, 1995. **62**(Copyright 1996, IEE): p. 11-26.
18. Chum, O. and J. Matas. *Geometric Hashing with Local Affine Frames*. in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. 2006.
19. Chum, O., M. Perdoch, and J. Matas. *Geometric min-Hashing: Finding a (thick) needle in a haystack*. in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. 2009.
20. Broder, A. Z. *On the resemblance and containment of documents*. in *Proceedings Compression and Complexity of SEQUENCES 1997, 11-13 June 1997*. 1998. Los Alamitos, CA, USA: IEEE Comput. Soc.
21. Andoni, Alexandr and Piotr Indyk, *Efficient algorithms for substring near neighbor problem*, in *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*2006, ACM: Miami, Florida. p. 1203-1212.
22. Grauman, K. and T. Darrell. *Fast contour matching using approximate earth mover's distance*. in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. 2004.
23. Ke, Van, Rahul Sukthankar, and Larry Huston. *Efficient near-duplicate detection and sub-image retrieval*. in *ACM Multimedia 2004 - proceedings of the 12th ACM International Conference on Multimedia, October 10, 2004 - October 16, 2004*. 2004. New York, NY, United states: Association for Computing Machinery.
24. Georgescu, B., I. Shimshoni, and P. Meer. *Mean shift based clustering in high dimensions: a texture classification example*. in *ICCV 2003: 9th International Conference on Computer Vision, 13-16 Oct. 2003*. 2003. Los Alamitos, CA, USA: IEEE Comput. Soc.
25. Matei, B., Shan Ying, H. S. Sawhney, Tan Yi, R. Kumar, D. Huber, and M. Hebert, *Rapid object indexing using locality sensitive hashing and joint 3D-signature space estimation.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006. **28**(Copyright 2006, The Institution of Engineering and Technology): p. 1111-26.
26. Grauman, K. and T. Darrell. *Efficient image matching with distributions of local invariant features*. in *Proceedings. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 20-25 June 2005*. 2005. Los Alamitos, CA, USA: IEEE Comput. Soc.
27. Grauman, K. and T. Darrell. *Pyramid match hashing: sub-linear time indexing over partial correspondences*. in *CVPR '07. IEEE Conference on Computer Vision and Pattern Recognition, 18-23 June 2007*. 2007. Piscataway, NJ, USA: IEEE.
28. Zixiang, Kang, Ooi Wei Tsang, and Sun Qibin. *Hierarchical, non-uniform locality sensitive hashing and its application to video identification*. in *2004 IEEE International Conference on Multimedia and Expo (ICME), 27-30 June 2004*. 2004. Piscataway, NJ, USA: IEEE.

29. Qiang, Wang, Tang Xiaoou, and H. Shum. *Patch based blind image super resolution*. in *Proceedings. Tenth IEEE International Conference on Computer Vision, 17-21 Oct. 2005*. 2005. Los Alamitos, CA, USA: IEEE Comput. Soc.
30. Shakhnarovich, G., P. Viola, and T. Darrell. *Fast pose estimation with parameter-sensitive hashing*. in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. 2003.
31. Jain, Prateek, Brian Kulis, and Kristen Grauman. *Fast image search for learned metrics*. in *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR, June 23, 2008 - June 28, 2008*. 2008. Anchorage, AK, United states: Inst. of Elec. and Elec. Eng. Computer Society.
32. Nister, David and Henrik Stewenius. *Scalable recognition with a vocabulary tree*. in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006, June 17, 2006 - June 22, 2006*. 2006. New York, NY, United states: Institute of Electrical and Electronics Engineers Computer Society.
33. Schindler, G., M. Brown, and R. Szeliski. *City-scale location recognition*. in *CVPR '07. IEEE Conference on Computer Vision and Pattern Recognition, 18-23 June 2007*. 2007. Piscataway, NJ, USA: IEEE.
34. Mikolajczyk, K. and H. Uemura. *Action recognition with motion-appearance vocabulary forest*. in *2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 23-28 June 2008*. 2008. Piscataway, NJ, USA: IEEE.
35. Coughlan, J. M. and A. L. Yuille, *Manhattan world: orientation and outlier detection by Bayesian inference.* Neural Computation, 2003. **15**(Copyright 2004, IEE): p. 1063-88.
36. *Google Earth*. Available from: http://www.google.com/earth/index.html.
37. Marr, David, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*1982: Henry Holt and Co., Inc.
38. Schindler, Grant, Panchapagesan Krishnamurthy, and Frank Dellaert. *Line-based structure from motion for urban environments*. in *3rd International Symposium on 3D Data Processing, Visualization, and Transmission, 3DPVT 2006, June 14, 2006 - June 16, 2006*. 2007. Chapel Hill, NC, United states: Inst. of Elec. and Elec. Eng. Computer Society.
39. Grauman, K. and T. Darrell. *The pyramid match kernel: discriminative classification with sets of image features*. in *Proceedings. Tenth IEEE International Conference on Computer Vision, 17-21 Oct. 2005*. 2005. Los Alamitos, CA, USA: IEEE Comput. Soc.
40. Lowe, D. G. *Object recognition from local scale-invariant features*. in *Proceedings of the Seventh IEEE International Conference on Computer Vision, 20-27 Sept. 1999*. 1999. Los Alamitos, CA, USA: IEEE Comput. Soc.
41. Papageorgiou, C. P., M. Oren, and T. Poggio. *A general framework for object detection*. in *Proceedings of IEEE 6th International Conference on Computer Vision, 4-7 Jan. 1998*. 1998. New Delhi, India: Narosa Publishing House.