A GRAPH MODEL FOR SCENE BASED IMAGE ANALYSIS AND CLASSIFICATION USING EPIPOLAR GEOMETRY

by

SIVABALAN MUTHUKUMAR, B.E.

A THESIS

IN

COMPUTER SCIENCE

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for
the Degree of

MASTER OF SCIENCE

Approved

Eric David Sinzinger
Chairperson of the Committee

Gopal D. Lakhani

Hector Juan Hernandez

Accepted

John Borrelli
Dean of the Graduate School

May, 2005

ACKNOWLEDGEMENTS

Special thanks to Dr. Eric Sinzinger for his guidance and patience during the course of this research. I would also like to thank Dr. Gopal Lakhani and Dr. Hector Hernandez for their reviews and guidance.

I would also like to thank my fellow graduate students in the Data Representation and Image Learning Lab for their support and encouragement. The couple of years that I spent in this lab was a wonderful experience.

I thank my parents for their continued support and the belief they had in me throughout my life. I would also like to thank my friends for their support and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ABSTRACT

This thesis presents a system that analyzes a collection of images and generates a model which describes the relationship between them. The main focus is to develop a system which can provide answers to the following two questions: Do the images match? Are they part of the same scene ? The answers to these questions can be used to classify the image collection into distinct groups. This transformation is achieved in three stages of processing. The firstis concerned with detection and extraction of features from images. The second stage focuses on matching the extracted features and the determination of the epipolar geometry. The final stage involves using the results obtained from the previous stage to develop a graph model. The main aim is to efficiently capture and represent the relationship between the images using this model and provide answers to the questions described above in a simple and effective manner. The goal is to classify the image collection into distinct group based on scene analysis by using the answers provided by the model regarding the nature of the relationship between the various images that are part of the given collection.

CHAPTER I
INTRODUCTION

The proposed work is an attempt to develop a robust and efficient system that can match and classify images. The main focus is to determine image connections in a fast and accurate manner. This can analyze scenes and perform an efficient image classification. The system will represent these results in a graph based model and use it to provide answers regarding the relationship existing between images.

The major part of this work is based on the epipolar geometry, related techniques like feature detection and matching which helps to determine the epipolar geometry. Epipolar Geometry is the intrinsic projective geometry of two views. It captures all the internal information constrained in two images.It is also independent of scene structure and its determination provides important information regarding the geometry between two views. This determination of epipolar geometry depends on detecting features across images and matching them using suitable feature selection and matching algorithms. The system also uses statistical measures for data analysis. Various concepts from graph theory are used to generate a graph which is representative of the geometry between the images.

The workings of the system can be grouped into three stages depending on their function. The first is the feature detection and extraction process. The second stage describes methods for matching the extracted features and using it to determine the geometry between the two views. a model for data representation and analysis of extracted data. The last stage describes the process of constructing a graph model. It includes a sequence of steps that deals with data representation, analysis and estimation of additional parameters using the available data. This is then used to determine the connectivity information between the images.

The function of the system can be summarized as transforming a collection of images into a graph that identifies images that are part of the same scene. The main application of the system is its ability to classify a collection of image into distinct groups. Hence this system could be used to obtain simple information like determining image connections and the connection strength. But the main motivation is its utility to applications involving scene based image classification like image based modeling and rendering. This system, for instance, could easily be adapted for developing an image based virtual tour. For example, a collection of

images of various locations in the campus, could be classified into distinct groups of different campus sections. It could also be used as a base system for other vision applications like scene modeling or image based rendering.

CHAPTER II

RELATED WORK

2.1 Introduction

The task of image classification and grouping is an important part of many applications. It can be defined as the process of identifying and descriping properties of image features and using it to classify the images into related groups. A speific class image classification methods focusses on identifying image matches. These methods can be used in many applications that require this kind of image classification. Some of the application domains include Image databases and Image based rendering techniques.

2.2 Image Databases

Image Databases can be described as a computer system in which the images are kept in an organized form. It facilitates organized storage and retrieval of digital images. The process of developing an image database could be described in the following sequence of steps:

1. Collecting images from various sources.

2. Converting the images into a digital format like JPEG, GIF or other image formats.

3. Adding metadata information like size, data, creator and description.

Image Databases can be used to store any kind of digital media, from slides to digital images. But they are commonly used to store images related to a specific category. Some of the popular image databases categories are medical images, astronomical images, historical images, art images and geographic images. These databases refer to a larger collection of images and are very generic in nature. These groups may be subdivided into further levels in order to facilitate efficient storage and retrieval. For instance, the art images may be categorized into different groups based on the time period. These may further be classified into subgroups based on the artist. The magnitude of these image databases may vary widely depending on the application they are intended for. For example, there are small sized image databases like campus image databases. These databases comprise of images from different locations in the campus.

Query by Image Content (QBIC)

An image retrieval system is a system that is used to browse, search and retrieve images from an image database. It is usually based on a textual description of the images in the database. The database is usually a metedata based system and requires manual annotation of each image in the database. This is impractical for large database, time-consuming and highly ambiguous.

A Query by Image Content (QBIC), also known as Content based Image retrieval (CBIR) is the application of computer vision to the problem of image retrieval. A QBIC system can be used to query large image databases based on visual image content, using properties such as color percentages, color layout, and textures occurring in the images. These queries uses virtual parts of image for comparison. Hence the images can be matched in terms of colors, textures or their positions without the need for textual description. An efficient method for this approach can be found in Faloutsos et al. (1994). A detailed description of various QBIC systems can be found in Smith and Chang (1996); Ma and Manjunath (1999). A retrieval technique using category based techniques for image retrieval could be found in Newsam et al. (2001).A comparison of various techniques for content based image retrieval system could be found in Laaksonen and Oja (2001).

Blobworld is one of the well known systems for content based image retrieval. The system functions by segmenting the image into regions which corresponds to different objects in the image. The images can then be queried by giving a description of the objects that they contain. The features that are used for querying includes color, texture, location and shape of the regions. The blobworld system can be used to classify images into different categories based on the objects in the image. More information about this image retrieval system can be found in Carson et al. (1999).

2.3 Image based Modelling and Rendering(IBMR)

Image-based rendering (IBR) refers loosely to techniques that generate new images from other images rather than from geometric primitives. A closely related concept is Image-based Modeling (IBM), which denotes any sampled representation of a 3D scene. Image-based modeling and rendering differs from traditional graphics in that both the geometry and appearance of the scene are derived from real photographs. The techniques often allow for

shorter modeling times, faster rendering speeds, and unprecedented levels of photorealism. Image based Modelling and Rendering (IBMR) is about leveraging the ease with which images can be taken and their power to communicate. Though a single image provides a lot of information about the scene's structure and appearance, it still remains a static frozen image. This has major limitations since it loses the ability to interact with the scene or to modify its structure. The IBMR approach transcends this limitations by using images from multiple view points and using it to generate models of the scene. Some well known IBR applications are the CityWalk project at MIT and the Campanile Movie project at University of Berkeley.

Paul Debevec's Campanile Movie : The Campanile Movie project by Paul Debevec was the first major breakthrough in the field of Image based modelling. The main aim of the project was to illustrate the power of IBMR approach and describe methods for recovering three dimensional models from photographs. Another important feature is that no real-time lighting is used. Instead the texture from the photographs are used to create the lighting effects.

The images used for the movie were mostly obtained by photographing using a kite. These along with the other images taken from the ground shots were then combined using registration techniques to recover the three dimensional model. The models were obtained using both automated tools and human assistance. Some related publications include Debevec (1996); Debevec et al. (1996).

Seth Teller's CityWalk Project: The CityWalk project is another project that illustrates the power and flexibility of taking an image based approach for rendering and modelling. The main aim was to develop a textured geometric model of an urban environment. The final model included a geometric representation of all the structures and its features that are part of that environment. The system was mostly automated but does require some human assistance for image acquisition.

A large collection of images taken from different viewpoints are used. A device was developed to acquire images and annotate it with navigation information using a global positioning system. These images were then registered using a common coordinate system by recovering the information about the camera poses. A detailed description about this can be found in Jethwa et al. (2003); Teller (1998).

CHAPTER III
BACKGROUND

3.1 Overview

The proposed system here is based on the concepts of Feature Selection, Matching and Epipolar Geometry. They describe methods and algorithms to analyze images and detect necessary parameters required by the system. These include detecting image features, matching them and estimating the fundamental matrix based on the matches.

3.2 Image Formation

3.2.1 Overview

The image formation model describes how points in space project onto the image plane. This process can essentially be reduced to tracing rays from points on objects to pixels on the image plane. A precise correspondence can be obtained between the points in 3-D space and their projected images in a 2-D image plane by using a mathematical model that accounts for the following three types of transformations:

1. Coordinate transformations between the camera frame and the world frame,
2. Projection of the 3-D coordinates onto the 2-D image coordinates,
3. Coordinate transformation between different image coordinate frames.

A detailed description of the image formation process can be found in Ma et al. (2004). A brief description of cameras and camera models can be found in (David A. Forsyth, 2003, pages 3-20) and in (Emanuele Trucco, 1998, pages 15-40).

3.2.2 Transformations

Coordinate transformation : Consider a generic point p, with coordinates $X_0 = [X_0, Y_0, Z_0]^T \in R^3$ relative to the world reference frame. The coordinates $X = [X, Y, Z]^T$ of p relative to the camera frame are given by the rigid-transformation $g = (R, T)$ of X_0:

$$X = gX_0 = RX_0 + T \in R^3 \tag{3.1}$$

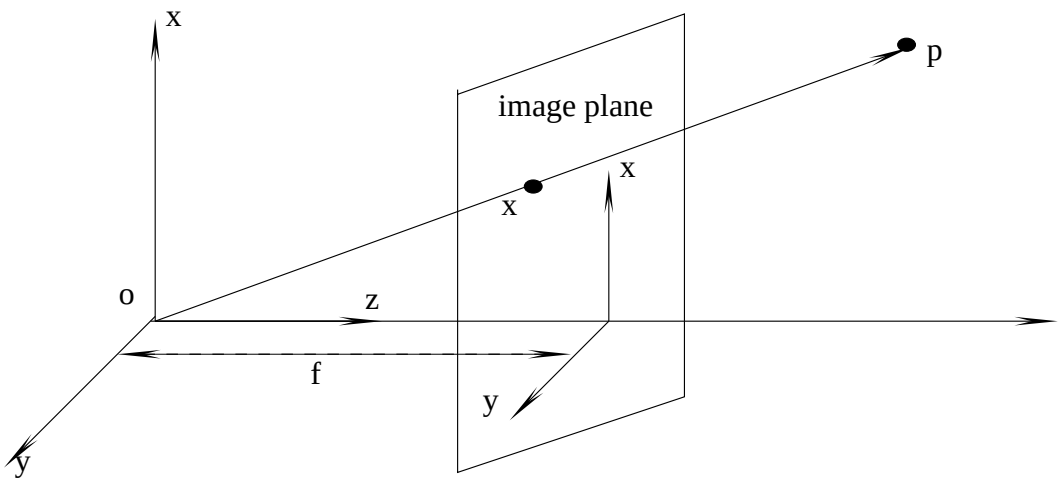


Figure 3.1: Pinhole Imaging Model

where R represents the rotation matrix and T represents the translation vector.

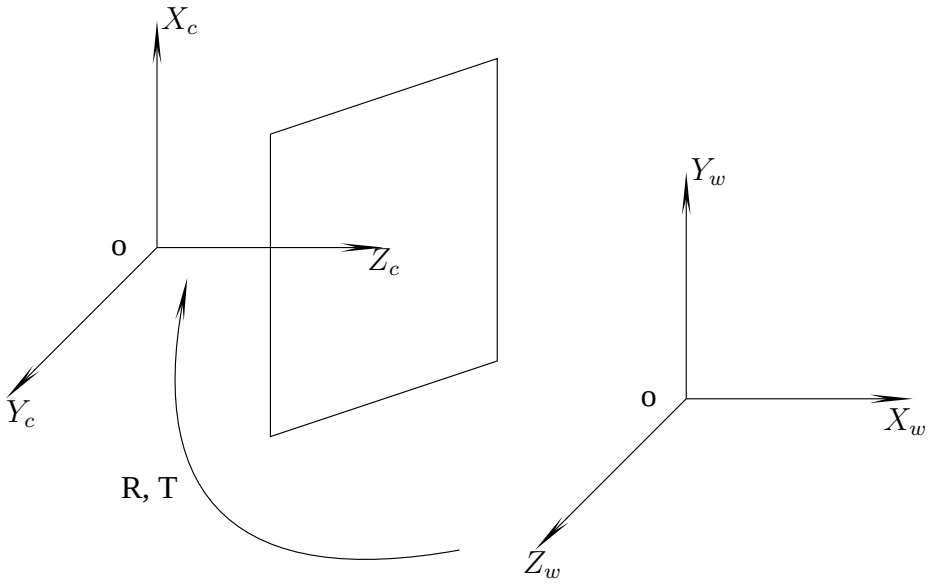


Figure 3.2: Euclidean transformation

Projection transformation : This describes the transformation from 3-D coordinates to the 2-D image coordinates. This transformation can be obtained by adopting the frontal pinhole camera model. The homogeneous image coordinates x and the spatial coordinates X of a

point p, with respect to the camera frame, are related by

$$\lambda x = K_f \Pi_0 X \tag{3.2}$$

where

$$K_f = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \in R^{3\times 3}$$

$$\Pi_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \in R^{3\times 4}.$$

This implies that the image x differs from the actual 3-D coordinates of the point by an unknown depth scale $\lambda \in R_+$. The matrix Π_0 is referred to as the "standard projection matrix". If the focal length f is known and can be normalized to 1, the transformations can be represented as Euclidean transformation g followed by a standard projection Π_0, i.e.

$$\lambda x = \Pi_0 X = \Pi_0 g X_0. \tag{3.3}$$

Camera to image transformation : The equation obtained in equation(3.2) is specified relative to the "canonical retinal frame". This is a specific type of reference frame which is centered at the optical center with one axis aligned with the optical axis. But in practice, the measurements are obtained in terms of pixels with the origin of the image coordinate frame in the upper-left corner of the image. Hence a relationship should be established between the retinal plane coordinate frame and the pixel array. This transformation is achieved by specifying the various parameters of the camera such as the focal length f, the scaling factors $s_x, s_y \, and \, s_\theta$, and the center offsets o_x, o_y. The projection model obtained in the previous section can be combined with scaling and translation to yield a more realistic model of transformation between the homogeneous coordinates of a 3-D point relative to the camera frame $x = [x, y, 1]^T$ and the homogeneous coordinates of its image expressed in terms of pixels $x' = [x', y', 1]^T$. The equation (3.2) can now be rewritten by incorporating the above

mentioned parameters.

$$\lambda x' = K_s K_f \Pi_0 X = K \Pi_0 X \tag{3.4}$$

where K represents the "intrinsic parameter matrix".

This intrinsic parameter matrix, denoted by K, is an upper triangular 3×3 matrix. It collects all the information that are intrinsic to the camera, and is therefore called the intrinsic parameter matrix or the calibration matrix. It has the following form:

$$K = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

The entries of K have the following geometric interpretation:

- o_x:x-coordinate of the principal point in pixels,
- o_y:y-coordinate of the principal point in pixels,
- $fs_x = \alpha_x$:size of unit length in horizontal pixels,
- $fs_y = \alpha_y$:size of unit length in vertical pixels,
- α_x / α_y:aspect ration σ,
- fs_θ:skew of the pixel, often close to zero.

3.2.3 Summary

The geometric relationship between a point of coordinates $X_0 = [X_0, Y_0, Z_0, 1]^T$ relative to the world frame and its corresponding image coordinates $x' = [x', y', 1]^T$ depends on the rigid-body motion (R,T) between the world frame and the camera frame, an ideal projection Π_0, and the camera intrinsic parameters K. The overall model for image formation can be

captured by the following equation:

$$\lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}$$

In matrix form, this can be written as

$$\begin{aligned} \lambda x' &= K\Pi_0 X = K\Pi_0 g X_0 \\ &= [KR, KT] X_0. \end{aligned}$$

The 3×4 matrix $K\Pi_0 g = [KR, KT]$, is known as the general projection matrix Π. Hence the above equation can be written as

$$\lambda x' = \Pi X = K\Pi_0 g X_0. \tag{3.5}$$

3.3 Image Features

Image Features typically can either be a global property of an image or a part of an image with some special properties known as the local feature. The features that are considered in this system are the local feature because they provide better means to track across multiple images and all the algorithms that are used assume that information about the local image features are available.

Based on the above discussion, Image Features can be defined as local, meaningful, detectable parts of the image that usually correspond to interesting elements of the scene. Meaningful and Detectable implies the fact that features are associated with interesting elements in the scene via the image formation process and there should be location algorithms that will be able to specify the position and other essential properties of the feature. A detailed description of various image features and algorithms for extracting them can be found in (Emanuele Trucco, 1998, pages 67-120). Some of the image features that are widely used in computer vision systems are as follows:

1. Edges are pixels at or around which the image values undergo a sharp variation. It

represents the changes that occurs in either the horizontal or the vertical direction. A widely used method is the Canny edge detector J.Canny (1986).

2. Corners are features that describe the intersection of image lines or a change in the pattern intensities. It represents a significant change in both horizontal and the vertical direction.

3.4 Feature Selection

Feature Selection involves selecting candidate features in the image, based on some criteria using various detection algorithms. The sequence of operations in most computer vision system begins by selecting some features in the input images depending on the requirements. The proposed system also begins with feature selection in one or more images, in preparation for matching them across multiple views. Since the focus is on feature matching, selecting corner features is the most viable option.

Corner features : A corner feature is the virtual intersection of local edges within a window. They can be identified by analyzing the spatial image gradient over the neighborhood W of the image point. The region W must have nontrivial gradients along two independent direction in order for a corner point to exist in the region.

Image Gradient : The description of these corner features relies on knowing the gradient of the image. Hence the image gradient $\nabla = [I_x, I_y]^T$ should be computed in an accurate and robust manner.

3.4.1 Selection process

A corner feature could be regarded as the intersection of all the edges inside the window W. The existence of a corner point $x = [x, y]^T$ means that over the window $W(x)$, the following matrix denoted by G remains nonsingular.

$$G(x) = \begin{bmatrix} \sum I_x^2 & \sum I_x.I_y \\ \sum I_x.I_y & \sum I_y^2 \end{bmatrix} \in R^{2x2} \tag{3.6}$$

A algorithm for detecting corner features using this is described in Algorithm 1. More information about this can be found in Tomasi and Kanade (1991).

Algorithm 1 Feature Selection

INPUT: An image I,two parameters: threshold on σ_2, τ, and size of window $W = 2N + 1$.

Compute the image gradient $\nabla = [I_x, I_y]^T$.
for each image point p **do**
    Form the matrix G of equation (3.6) over a neighborhood W of p.
    Compute σ_2, the smaller eigen value of G.
    if $\sigma_2 > \tau$ **then**
        Mark the pixel p as a corner point and store its coordinates into a list, L
    end if
end for
Sort the list L in descending order of σ_2.
Scan the sorted list from top to bottom.
for each point in L **do**
    Delete all points appearing further in the list which belong to neighborhood of p.
end for

OUTPUT: List of corner features and its coordinates.

3.4.2 Remarks

- There is no optimal criterion for the estimation of the optimal size of the neighborhood. Experience with this method indicates that the optimal value that can be used for N is between 2 and 10. In case of corners that value of N is linked to the location of the corner point within the neighborhood. When the value of N is large the point tends to move away from the center.

- The threshold, τ , can be estimated from the histogram or by repeated experiments with different values.

- At corner points, the intensity surface has two well-pronounced, distinctive directions, associated to the eigen values σ_1 and σ_2 of G, both significantly larger than 0.

- If σ_2, the smaller eigen value of G, is above a specified threshold τ, then G is invertible and the point x is a feature point.

- If both singular values of G are close to zero, it implies that the window is a region with almost constant brightness.

- If any one of the singular values is close to zero, it implies the brightness varies in only one direction.

- In the last two cases, the point cannot be localized or matched in another image.

3.4.3 Variation: Harris Corner Detector

A variation to the above algorithm is the Harris corner detector. The main idea here is to threshold the quantity

$$C(G) = det(G) + k \times trace(G) \tag{3.7}$$

where $k \in R$ is a small scalar, and different choices of k may result in favoring gradient variation in one or both directions. If k is small, both eigen values should be large enough to make $C(G) > \tau$, thereby favoring the corner feature. More information about this can be found in Harris and Stephens (1988). A computational approach for detecting corners and various methods to improve the accuracy can be found in Deriche and Giraudon (1993).

Feature Descriptor : The feature descriptor for a corner feature should describe the location of the corner as well as its strength. The location is given in pixel coordinates while the strength is a positive real value based on the quantity $C(G)$ or the minimum eigen value σ_2.

3.5 Feature Matching

Feature Matching involves matching the selected point features from an image pair. There are many different methods that can be used to match these selected point features. Many of these methods assume that the following two conditions are satisfied.

1. Most scene points are visible from both viewpoints.

2. Corresponding image regions are similar.

These assumptions hold for systems in which the fixation point from the cameras is much larger than the baseline. The main aim in this step is to find a feasible solution for the correspondence problem. A detailed description about the matching process can be found in (Ma et al., 2004, pages 75-105).

Correspondence Problem : The correspondence problem consists in establishing which point in one image corresponds to which point in the other image, in the sense that the two points being considered are the projection of the same point in space. It can essentially be viewed as a search problem: given an element in the left image, find the corresponding element in the right image.

This involves two decisions:

- determining the right elements to match, and
- deciding the best similarity measure to adopt.

3.5.1 Image Deformation

Image Deformation is the transformation undergone by the image between two views. This transformation can either be a change in intensity or a change in structure or a combination of both. This deformation can either be interpreted as a local deformation or a global deformation.

Local image deformation describes the transformation of the matching window around the neighborhood of the point while global image deformation describes the transformation over the entire image. The deformation undergone by the entire image cannot be easily captured by a simple transformation. This results in two opposing strategies: the first is to choose a complex transformation that captures the changes undergone by the entire image, the other one is to choose a simple transformation, and restrict the attention to only those regions in the image whose motion can be captured.

General Deformation model : An image I_1 can be represented as function on a compact two-dimensional region Ω taking irradiance values in the positive reals.

$$I_1 : \Omega \subset R^2 \rightarrow R_+; x \mapsto I_1(x)$$

Similarly, a different image of the same scene, I_2 can be represented as

$$I_2 : \Omega \subset R^2 \rightarrow R_+; x \mapsto I_2(x).$$

Correspondence: Consider two images, I_1 and I_2, of the same scene. Let x_1 and x_1 be points on the image. Therefore, if x_1 and x_1 are the two images of the same point p in the two view, respectively, then the following equation must hold good.

$$I_1(x_1) = I_2(x_2) \tag{3.8}$$

This implies that the two points must have the same value, i.e. they should have the same color. Under this assumption, the correspondence problem consists of verifying that the two points x_1 and x_1 are indeed images of the same 3-D point.

Model: A model for the deformation between two images of the same scene is given by an image matching constraint

$$I_1(x_1) = I_2(h(x_2)). \tag{3.9}$$

This equation is also knows as the *brightness constancy constraint*, since it expresses the fact that given a point on an image, there exists a different point in another image that has the same brightness. The function h describes the transformation of the domain, or "image motion". It can be represented as

$$h(x) = x + \Delta(x)(X). \tag{3.10}$$

3.5.2 Local deformation models

Local deformation models concentrates on choosing a class of simple transformations, and then focusing on specific regions of the image which can be modeled as undergoing the chosen transformation. The problem associated with the global model, is that the transformation undergone by the entire image is infinite-dimensional and finding it amounts to inferring the 3-D structure of the image. It also requires a highly complex transformation to model an image deformation over the entire image. Hence by using a local deformation model, the problems associated with a global model are eliminated. A description of image deformation and feature matching can be found in (Ma et al., 2004, pages 77-92).

In general, in the local deformation model, each point is associated with a support window and the value of each point in the window. Both the window shape and the image values undergo *transformations* as a consequence of a change in viewpoint. Such transformation occur in the domain of the image, the subwindow $W(x)$ around x, and in the intensity values $I(\tilde{x}), \tilde{x} \in W(x)$.

3.5.2.1 Transformations of the image domain

Translational motion model : This is the simplest model in which each point in the window undergoes the same motion, i.e. $\Delta x = constant$. This implies that there is no dependency on

X, refer 3.10. This model can be represented as

$$h(\tilde{x}) = \tilde{x} + \Delta x,\ \forall \tilde{x} \in W(x),\ where\, \Delta x \in R^2. \tag{3.11}$$

This model is valid only for scenes that are flat and parallel to the image plane, and moving parallel it. Hence this model is valid locally in space, as in small windows, and in time, as in adjacent frames or small camera motion. The advantages of the translational model are its simplicity and efficiency. Hence this is used in most feature matching or tracking algorithms resulting in a very efficient algorithm, which is presented in Algorithm 2.

Affine motion model In the affine motion model, the points in the window $W(x)$ do not undergo the same motion as in translation motion model, but, instead the motion depends linearly on its location plus a constant offset. This model can be represented as

$$h(\tilde{x}) = A\tilde{x} + d,\ \forall \tilde{x} \in W(x),\ where\, A \in R^{2\times 2},\ d \in R^2. \tag{3.12}$$

This model is a good approximation for scenes that have small planar patches moving parallel to the image plane undergoing arbitrary translation and rotation about the optical axis, and modest rotation about an axis orthogonal about it. The advantages of using this model is that it represents a convenient trade off between simplicity and flexibility and provides a richer model when compared to the translational motion model.

3.5.2.2 Transformation of the intensity domain

The derivations in the previous section assumes that each image of the original 3-D point results in the same irradiance in both views.But in practice, this is unrealistic due to many different factors. Thus this additional complication resulting in different intensity values can be modeled by using a additive noise term n to the general deformation model. This additive term represents all the sources of uncertainty put together into a single term. Therefore, equation (3.9) must be modified to account for changes in the intensity value, in addition to the image domain deformation. This can be represented as

$$I_1(x_1) = I_2(h(x_2)) + n(h(x_1)). \tag{3.13}$$

3.5.3 Matching Process

3.5.3.1 Overview

The matching process to solve the correspondence problem considers the local photometric information of the support window $W(x)$ around the point x. Instead of considering equation (3.8) in terms of points alone,it is now defined using regions instead of points. This is done by integrating each side on a windows $W(x)$ around each point x, and using this to define the correspondence at x. The process assumes that the translation motion model is used. The formulation of this process is due to B.D.Lucas and T.Kanade (1981) and later refined by Tomasi and Kanade (1992).

3.5.3.2 Formulation

The correspondence problem can now be formulated as the solution to the optimization problem. It begins by choosing a class of transformations that can be applied, and then finding the particular transformation $\hat{h}$ that minimizes the effect the additive noise term n, defined in equation 3.13.This can be written as

$$\begin{aligned} \hat{h} &= arg\, min_h \sum_{\tilde{x} \in W(x)} \|n(\tilde{x})\|^2 \\ &= arg\, min_h \sum_{\tilde{x} \in W(x)} \|I_1(\tilde{x}) - I_2(h(\tilde{x}))\|^2. \end{aligned} \tag{3.14}$$

3.5.3.3 Discrepancy criteria

The corresponding element is given by the transformation $\hat{h}$ that minimizes the additive noise term n. This minimization of performed according to some criterion. There are various choices of discrepancy criteria which could be used for this transformation. Some of the widely adopted choices are

1. Sum of squared differences (SSD)

2. Normalized cross-correlation (NCC)

The criteria that is preferred for matching template regions is the NCC. This is because of the fact that it more tolerant than the SSD when there are scalings and shifts in image intensities.

NCC : Given two nonuniform image regions $I_1(\tilde{x})$ and $I_2(\tilde{x})$, with $\tilde{x} \in W(x)$ and $N = |W(x)|$ (the number of pixels in the window), the normalized cross-correlation id defined as

$$NCC(h) = \frac{\sum_{W(x)}(I_1(\tilde{x}) - \bar{I_1}) - (I_1(\tilde{x}) - \bar{I_1})}{\sqrt{\sum_{W(x)}\left(I_1(\tilde{x}) - \bar{I_1}\right)^2 - \sum_{W(x)}\left(I_2(\tilde{x}) - \bar{I_2}\right)^2}} \tag{3.15}$$

where $\bar{I_1}$ and $\bar{I_2}$ are the main intensities:

$$\begin{aligned} \bar{I_1} &= \frac{1}{N}\sum_{W(x)} I_1(\tilde{x}) \\ \bar{I_2} &= \frac{1}{N}\sum_{W(x)} I_1(h(\tilde{x})) \end{aligned}$$

The normalized cross-correlation value always ranges between -1 and +1, irrespective of the window size. When the NCC value is 1, it implies that the image regions match perfectly. A method for increasing the speed of this method is described in Lewis (1995).

Algorithm 2 Basic Feature Matching

INPUT: Point features in two views I_1, I_2. Parameter: Threshold τ_e.

1: **for** each point feature x_1 in I_1 **do**
2: find the feature $x_2 = x_1 + d$ which maximizes the NCC similarity score between $W(x_1)$ and $W(x_2)$.
3: **if** (NCC Score between $x_1, x_2 > \tau_3$) **then**
4: mark the features with coordinates x_1, x_2 as a match
5: **end if**
6: **end for**

OUTPUT: List of matching features and their NCC score.

3.5.4 Remarks

1. The window size of the neighborhood is typically chosen between 5×5 and 21×21.

2. The window size is directly proportional to the computational cost and to the number of outliers. But it results in an increased robustness and a better match between the features.

3. This algorithm is realistic for computing correspondence between a few distinctive features across the views.

Feature Descriptor : The feature descriptor for a feature match should describe the feature location in each image and the match strength. The location is given in pixel coordinates, one for each image. The match strength is a real value between -1 and +1 based on the NCC score.

3.6 Epipolar Geometry

Epipolar Geometry is the underlying geometry between two images of the same scene. Two images of a single scene or object are related by this geometry which captures all the geometric information contained in two images. The determination of the epipolar geometry is a very important step for determining the relationship between the images.

3.6.1 Notation

The representation of the epipolar geometry is shown in Figure. It consists of two pinhole cameras, their projection centers, o_1 and o_2, and image planes, π_1 and π_2. The focal lengths are denoted by f_1 and f_2. Each camera identifies a 3-D reference frame, the origin of which coincides with the projection center, and the Z-axis with the optical axis. The points $X_1 = [X_1, Y_1, Z_1] \in R^3$ and $X_2 = [X_2, Y_2, Z_2] \in R^3$ refer to the same 3-D point p in space, relative to the left and right camera reference frames respectively. The points x_1,$x_2 \in R^3$ refer to the projections of the point p onto the left and the right image plane respectively, and are expressed in the corresponding reference frame. The following are the epipolar geometric entities:

1. The epipolar plane is the plane (o_1, o_2, p) determined by the two centers of projection o_1, o_2 and the point p. The plane thus formed is associated with the camera configuration and point p. There is one epipolar plane for each point and is denoted by π_p.

2. The epipole is the projection of one camera center onto the image plane of the other camera frame is called an epipole. It is denoted by e_1, e_2 corresponding to left and right epipoles respectively. The projection may occur outside the physical boundary of the imaging sensor. Consequently, the epipoles may lie outside the image planes.

3. The epipolar line is the intersection of the epipolar plane of $p(\pi_p)$ with the image planes. It is denoted by l_1, l_2.

Basics : The reference frames of the left and right cameras are related via the extrinsic parameters. The relationship between the 3-D coordinates of a point in the world coordinate frame and the camera frame can be expressed as a rigid body transformation in 3-D space defined in space. The world frame can be assumed to be one of the cameras. The other camera is then positioned and oriented according to a Euclidean transformation $g = (R, T)$, where $T = (o_2 - o_1)$ defines a translation vector, and R defines a rotation matrix. This can be represented as

$$X_2 = RX_1 + T. \tag{3.16}$$

This equation is referred to as the rigid-body motion equation.

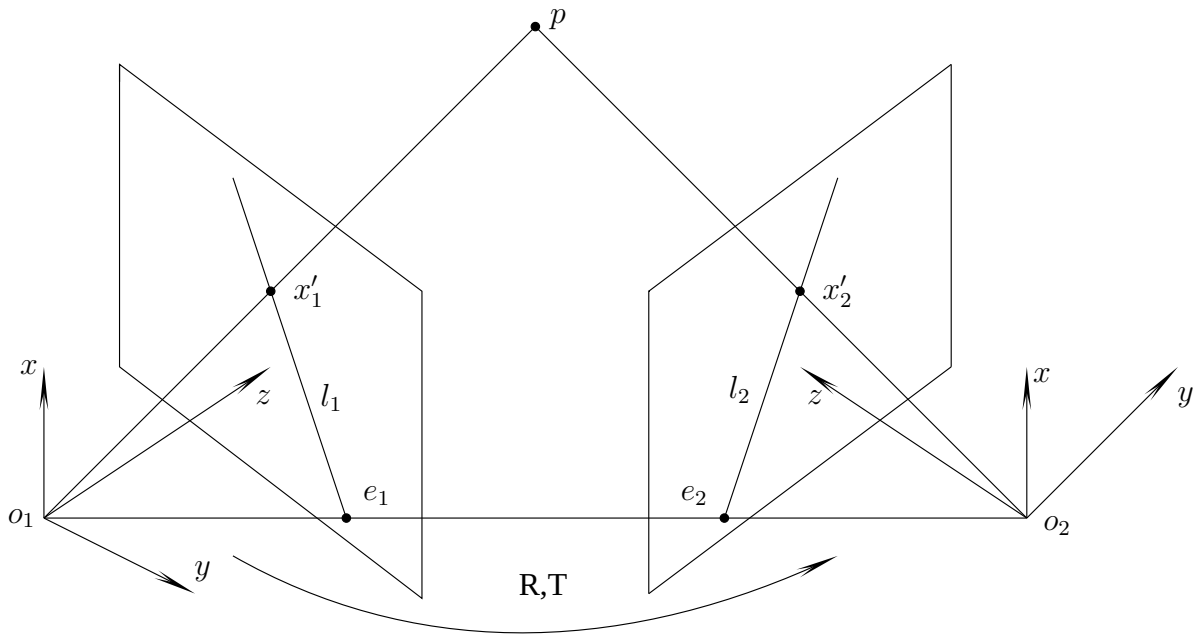


Figure 3.3: Epipolar Geometry

Epipolar Constraint : Consider the point p and its projection x_1, x_2. The point x_2 has to lie on the image, in the second camera, of the optical ray (o_1, p) of the first camera. This constraint is the epipolar constraint: For each point x_1 in the first image, its corresponding point x_2 in the second images lies on its epipolar line l_2. Similarly, for a given point x_2 in the second images, its corresponding point x_1 in the first image lies on its epipolar line l_1. These two lines are called corresponding epipolar lines.

3.6.2 Essential Matrix

The epipolar geometry can be described by a 3×3 matrix called the essential matrix when information about the intrinsic parameters of the system are available. It represents a natural link between the epipolar constraint and the extrinsic parameters of the stereo system. The essential matrix is defined in terms of the camera coordinates. A detailed description on calibrated epipolar geometry and essential matrix is given in (Ma et al., 2004, pages 109-170).

Algebraic Derivation : Consider the points x_1, x_2 in Figure. Since $X_i = \lambda x_i$ (equation 3.3), the equation 3.16 can be rewritten in terms of the image coordinates x_i and the depths λ_i as

$$\lambda_2 x_2 = R\lambda_1 x_1 + T.$$

Premultiplying the preceding equation by $\hat{T}$ to obtain

$$\lambda_2 \hat{T} x_2 = \hat{T} R \lambda_1 x_1.$$

Since the vector $\hat{T}x_2 = T \times x_2$ is perpendicular to the vector x_2, the inner product between these two vector is zero.

$$\langle x_2, \hat{T} x_2 \rangle = 0$$
$$x_2{}^T \hat{T} x_2 = 0$$

Premultiplying the previous equation by $x_2{}^T$ to obtain

$$\lambda_2 x_2{}^T \hat{T} x_2 = x_2{}^T \hat{T} R \lambda_1 x_1.$$

This yields that the quantity $x_2{}^T \hat{T} R \lambda_1 x_1$ is zero. Since $\lambda_1 > 0$, this implies that

$$x_2{}^T \hat{T} R x_1 = 0. \tag{3.17}$$

This equation is the epipolar constraint equation in the case of calibrated geometry. The matrix

$$E = \hat{T} R \in R_{3\times 3}$$

in the above epipolar constraint equation (3.17) is called the essential matrix. The properties of the essential matrix are listed below.

1. It encodes information on the extrinsic parameters.
2. It has rank 2.
3. It has two equal nonzero singular values.

3.6.3 Fundamental Matrix

The epipolar geometry can be described by a 3×3 matrix, known as the fundamental matrix, when there is no prior information available on the stereo system, i.e. the intrinsic parameters of the system are not available. The fundamental matrix is defined in terms of the pixel coordinates. Hence it can be estimated from a number of point matches in pixel coordinates. A detailed description of fundamental matrix and its estimation is given in (Ma et al., 2004, pages 171-227). A description of various linear and linear methods for determining fundamental matrix and a study of stability issues could be found in Luong and Faugeras (1996). A review of various methods for determining the epipolar geometry and its uncertainty can be found in Zhang (1996). A new method for nonlinear estimation of the fundamental matrix can be found in Minimal (2004).

Algebraic Derivation : The algebraic derivation can be obtained by following the same procedure as the calibrated case, i.e. by eliminating the depth scales from the rigid-body motion equation (equation 3.16). The equation 3.16 can be rewritten in terms of the image coordinates x_i and the depths λ_i as

$$\lambda_2 x_2 = R\lambda_1 x_1 + T.$$

Multiplying both sides by the calibration matrix K,

$$\lambda_2 K x_2 = KR\lambda_1 x_1 + KT \Leftrightarrow \lambda_2 K x_2' = KRK^{-1}\lambda_1 x_1' + T \tag{3.18}$$

where $x' = Kx$ and $T' = KT$. Multiplying the above equation by $T' \times x_2' = \hat{T}' x_2'$,

$$x_2' \hat{T}' KRK^{-1} x_1' = 0. \tag{3.19}$$

This equation is the epipolar constraint equation for an uncalibrated system. The matrix

$$F = \hat{T}' K R K^{-1} \tag{3.20}$$

in the above constraint equation is the fundamental matrix F.

The epipolar constraint equation in the case of uncalibrated geometry is given by

$$x_2'^T F x_1' = 0 \tag{3.21}$$

The fundamental matrix maps a point x_1' in the first view to a vector $l_2 = F x_1'$ in the second view via $x_2'^T l_2 = 0$. The vector l_2 defines a line in the second view as a collection of image points x_2' that satisfy the equation $l_2^T x_2' = 0$.

3.6.4 Computation of F

3.6.4.1 The Eight-point Algorithm - Overview

The eight-point algorithm is the simplest algorithm available for computing F. The idea behind the eight-point algorithm is very simple. Each point correspondence between the two images gives a homogeneous linear equation for each of the nine entries of F forming a homogeneous linear system. If there are at least eight correspondences and they do not form a degenerate configuration, the nine entries of F can be determined as the nontrivial solution of the system. If there are more than 8 points, then the solution can be obtained by using SVD related techniques. This algorithm is due to H.C.Longuet-Higgins (1981). A description of this algorithm can be found in (Ma et al., 2004, pages 211-212) and in (Emanuele Trucco, 1998, pages 155-156).

3.6.4.2 Formulation

The fundamental matrix can be recovered from the detected point correspondences using linear techniques. The fundamental matrix F represented as

$$\begin{bmatrix} f_1 & f_4 & f_7 \\ f_2 & f_5 & f_8 \\ f_3 & f_6 & f_9 \end{bmatrix}$$

are stacked into a vector $F^s \in R^9$,

$$F^s = [f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9]^T \ \in \ R^9.$$

Since the epipolar constraint $x_2' F x_1' = 0$ is linear in the entries of F, it can be rewritten as

$$a^T F^s = 0 \tag{3.22}$$

where $a = x_2' \otimes x_1'$ is given by

$$a = [x_1'x_2', x_1'y_2', x_1'z_2', y_1'x_2', y_1'y_2', y_1'z_2', z_1'x_2', z_1'y_2', z_1'z_2']^T \ \in \ R^9. \tag{3.23}$$

Given a set of $n \geq 8$ corresponding points, a matrix of measurements $\chi = [a^1, a^2, \ldots, a^n]^T$ is first formed. F^s can then be obtained as the minimizing solution of least-squares objective function $\|\chi F^s\|^2$. This solution corresponds to the eigenvector associated with the smallest eigen value of $\chi^T\chi$, and can be found by using the SVD and setting F^s to be the right singular vector associated with the smallest singular value.

3.6.4.3 Normalization

The eight-point algorithm must be implemented with care to avoid numerical instabilities. The most important action is to normalize the coordinates of corresponding points so that the entries of χ are of comparable size. Since image coordinates x_1' and x_2' are measure in pixels, the individual entries of the matrix χ can vary between a few pixels to a few hundreds. These differences can make χ seriously ill-conditioned. Hence a transformation must be used to balance the coordinates. This can be done by transforming the points $\{x_1'^j\}_{j=1}^n$ by an affine matrix $H_1 \in R^{3\times 3}$ so that the resulting points have zero mean and unit variance. This normalization procedure to avoid numerical instabilities is due to R.I.Hartley (1995).

$$\tilde{x_i} = H_i x_i' = \begin{bmatrix} 1/\sigma_{xi} & 0 & -\mu_{xi}/\sigma_{xi} \\ 0 & 1/\sigma_{y_i} & -\mu_{y_i}/\sigma_{y_i} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} \tag{3.24}$$

where μ_{xi} is the average and σ_{xi} is the standard deviation. The above transformation H_1 transforms the pixel coordinates x'_i into normalized coordinates $\tilde{x'_i}$.

$$\begin{aligned} \mu_{xi} &= \frac{1}{n}\sum_{j=1}^{n}(x'_i)^j, \\ \sigma_{xi} &= \sqrt{\frac{1}{n}\sum_{j=1}^{n}[(x'_i)^j - \mu_{xi}]^2}. \end{aligned}$$

The details of this normalization method can be found in (Ma et al., 2004, pages 212-213).

Algorithm 3 Computing F - Eight-point algorithm

INPUT: Set of initial point correspondences expressed in pixel coordinates $({x'_1}^j, {x'_2}^j)$, $j = 1, 2, \ldots, n$,$n \geq 8$.

NORMALIZATION:

1: Normalize the image coordinates by $\tilde{x_1} = H_1 x'_1$ and $\tilde{x_2} = H_2 x'_2$, where H_1, H_2 are normalizing transformation as in equation (3.24).

FIRST APPROXIMATION:

2: Construct the system as in equation (3.23) from the transformed correspondences $\tilde{{x_1}}^j = [\tilde{{x_1}}^j, \tilde{{y_1}}^j, 1]^T$ and $\tilde{{x_2}}^j = [\tilde{{x_2}}^j, \tilde{{y_2}}^j, 1]^T$.

3: The entries of $\tilde{F}$ are the components of the column of V corresponding to the least singular value of χ.

4: The inverse normalizing transformations are applied to obtain $F = {H_2}^T \tilde{F} H_1$.

ENFORCE THE RANK-2 CONSTRAINT

5: Compute SVD of the recovered matrix F as $F = U_F diag\{\sigma_1, \sigma_2, \sigma_3\} {V_F}^T$. The rank-2 constraint is is imposed by letting $\sigma_3 = 0$. The fundamental matrix F is then reset as

$$F = U_F diag\{\sigma_1, \sigma_2, 0\} {V_F}^T$$

OUTPUT: Fundamental matrix F.

Algorithm 4 Epipoles Location

INPUT: Fundamental matrix F

1: Find the SVD of F, $F = UDV^T$.

2: e_1 is the column of V corresponding the null singular value.

3: e_2 is the column of U corresponding the null singular value.

OUTPUT: Epipoles e_1 and e_2.

Locating epipoles from F : The epipole, denoted by e, is the point where the baseline intersects the image plane in each view. The epipoles, e_1 and e_2, with respect to the first and the second view respectively, can be computed as the null space of the fundamental matrix F,

i.e. $Fe_1 = 0$. This is described in Algorithm 4. The details of this algorithm can be found in (Emanuele Trucco, 1998, pages 156-157).

Feature Descriptor : The feature descriptor for the fundamental matrix is a $3 \times 3 \in R^{3\times3}$. The epipoles are described using a matrix such that $e_1, e_2 \in R^3$.

CHAPTER IV
METHODOLOGY FOR GRAPH MODEL CONSTRUCTION

This chapter gives a detailed description of all the processes and the steps involved in developing the system that classifies the images into groups. It also sheds light on the various issues involved in developing such a system and a discussion on some of the ways to address them. The main aim is to analyze an image collection, extract feature information from those images and develop a model that can capture the connectivity information about the images. A pair of images are said to be connected if they overlap each other or if they are part of the same scene. There is no restriction on the type of camera that is used for capturing the images but there are some factors which are assumed in this system to help improve the classification. The assumptions are

- Feature Availability : This is an important assumption and implies that the images used should contain a sufficient amount of salient features.
- Feature Visibility : This assumption requires that images of the same scene should have a majority of their features visible across a minimum of two views of the scene.
- Image Availability : This implies that a sufficient number of images of a scene are available. This assumption, though not being a requirement for the system to function, enables it to generate a more accurate and robust model.

The goal is to analyze a collection of images and generate a model. The model should provide answers to the following questions in a simple and effective manner.

- Do the images match : This helps to identify images which overlap each other, i.e. they have a common region.
- Are they part of the same scene : This helps to identify images which may or may not overlap but are part of the same scene, i.e. they are related through other images of the scene.

This helps in identifying image matches and to classify the image collection into distinct scenes.

4.1 Feature Extraction

Feature extraction describes a method for extracting the salient features from the images. This is based on Algorithm 1 described in Section 3.4. The algorithm is slightly modified in order to improve some characteristics of the feature detection mechanism. Some of the modifications include increasing the speed of the detection process and to improve the distribution of features across the entire image. The estimation of corner measure is based on the Harris corner detector in Algorithm 1 and described in more detail in Harris and Stephens (1988). The image is first convolved using the Prewitt operator in both x and y directions to obtain I_x and I_y. Compute $I_{xy} = I_x.I_y$. The corner measure for each point is then given by

$$measure(M) = (I_x^2.I_y^2 - I_{xy}^2)/(I_x^2 + I_y^2). \tag{4.1}$$

The Prewitt operator is defined by eight convolution masks, with each mask corresponding to the eight possible compass directions. It approximates the first derivative and is described by a 3×3 matrix. The convolution results of the greatest magnitude give the gradient direction. The operator used for estimating the corner measure are h_1 and h_3.

$$\begin{aligned} h_1 &= \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \\ h_3 &= \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \end{aligned} \tag{4.2}$$

The features are selected in a distributed manner across the whole image. This helps in eliminating the feature localisation and thereby facilitating even distribution across the entire image. This is achieved by dividing the image into tiles. The features are sorted according to the corner measure M and the required number of features are selected from the tile provided they exceed the threshold τ. The process is described in the Figure (4.1).
Remarks :

1. The windows size W used is usually between 7×7 and 21×21. The size used here is

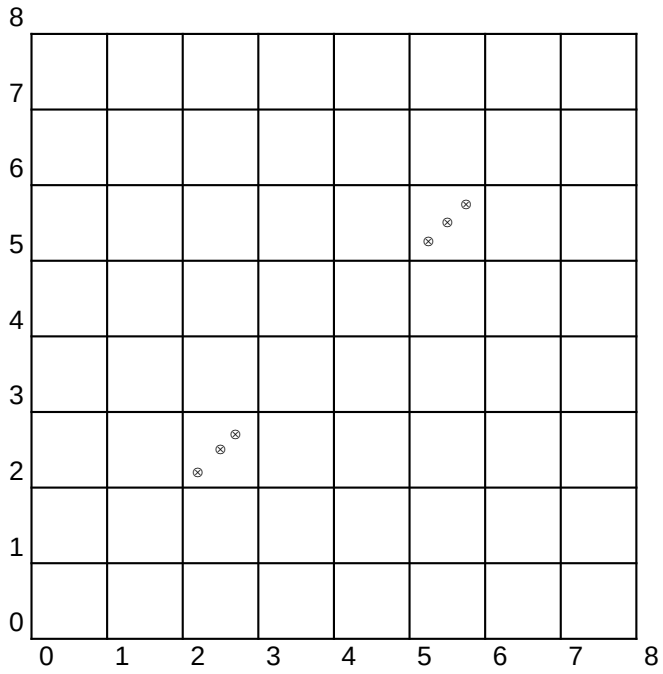


Figure 4.1: Point Feature Detection

13×13.

2. The threshold value (τ) used is 6.0.

3. The boundary size B is fixed at 10 pixels.

4. The number of tiles used is 8×8 giving a total of 64 tiles.

5. The maximum number of features per tile is 5.

The feature descriptor now requires an additional parameter to store the tile location in addition to the corner location and its strength. It is denoted by FD_{corner}. The feature descriptor is now described by the following data:

1. Location : Gives the (x, y) coordinates of the selected feature.

2. Strength : A measure of strength of the corner feature.

3. Tile Location: Gives the tile location (xT, yT) that the feature belongs. The values of xT and yT range from 0 to 7.

Algorithm 5 Point Feature Detection

INPUT: An image I,parameters: threshold(τ), Window Size(W), Boundary(B), Tilesize(hT,vT), Features per tile(nT).

Compute the image gradient $\nabla = [I_x, I_y]$ using the Prewitt operators h_1 and h_2 defined in equation(4.2).
Compute the corner measure matrix M defined in equation(4.1).
for each image point p inside the boundary B **do**
    if $measure(p) > \tau$ **then**
        Mark the pixel p as a corner feature and store its coordinates into a list, L
    end if
end for
Sort the list L in descending order of corner strength.
Scan the sorted list from top to bottom.
for each point in L **do**
    Determine the tile ($tilenum$) that the feature belongs to.
    Delete all points appearing further in the list which belong to neighborhood of p.
    if $features(tilenum) < nT$ **then**
        Retain the feature and add information about the tile location.
    end if
end for

OUTPUT: List of corner features and its coordinates.

4.2 Matching and Geometry

4.2.1 Exhaustive Feature Matching

Exhaustive feature matching describes a method based on the Algorithm 2 given in Section 3.5 to match the features detected during the point feature detection process. The matching technique used is exhaustive search over all the selected image feature across both images. The features selected for further processing are those that have a high degree of correlation between them. The window size W is 13×13. The threshold for the NCC score τ_e is 0.69. This is presented in Algorithm 6.

The feature descriptor for a match stores the matching features in the left and the right image and their NCC Score. It is denoted by FD_{match}. The description is given below:

1. Feature descriptor for left corner.
2. Feature descriptor for right corner.
3. NCC Score.

Algorithm 6 Exhaustive Feature Matching

INPUT: Point features in two views I_1, I_2. Parameter: Threshold τ_e.

for each point feature x_1 in I_1 **do**
    Search all the point features x_2 across I_2
    Compute the NCC score [NCC(x_1, x_2)]between their neighborhood W.
    if (NCC$(x_1, x_2) > \tau_e$) **then**
        mark the features with coordinates x_1, x_2 as a match and add it to a list L_m.
    end if
end for
Sort the list L in descending order of the NCC Score.
for each match m in L_m **do**
    Select the match with the highest NCC Score.
    Discard all the matches for that feature.
end for

OUTPUT: List of matching features and their NCC score.

4.2.2 Robust Computation of the Fundamental Matrix

A robust method is used to compute the fundamental matrix in an accurate manner by eliminating false matches obtained during feature matching. The algorithm used for actual computation of the fundamental matrix is the Eight Point Algorithm (Algorithm 3) described in Section (3.6.4). There are many robust methods available of which the most widely used are the Random Sample Consensus (RANSAC) and Lease Median of Squares(LMedS) approaches. The approach used here is the RANSAC method. A review of various techniques for determining the fundamental matrix can be found in Zhang (1996). A description of the RANSAC approach is given in (Ma et al., 2004, pages 388-389).

The idea behind RANSAC is to use a minimal number of points needed to estimate a model, and then count the number of points over the entire data set that is compatible with the estimated model. This approach is used to compute the fundamental matrix by using the minimum number of matches (8 in the case of Eight Point Algorithm used here). The quality of the matches is determined by the number of matches that is compatible with the computed fundamental matrix. The quantity used for minimization is the distance between the points $(\tilde{x}_1^j, F\tilde{x}_2^j)$ and the corresponding epipolar lines (l_2^j, l_1^j). The role of the two images are exchanged and the distance is calculated again using the transposed fundamental matrix. The distance measure obtained is given in the following equation (4.3).

$$d_j = d^2(\tilde{x}_2^j, F\tilde{x}_1^j) + d^2(\tilde{x}_1^j, F^T\tilde{x}_2^j) \tag{4.3}$$

The fundamental matrix F maps a point x in one view to a vector l in the other view. Using this, the following equation (4.4) is obtained.

$$\text{Let} \quad l_2^j = F\tilde{x}^j = [l_{21}, l_{22}, l_{23}]^T \quad \text{and} \quad \text{let} \quad l_1^j = F^T\tilde{x}^j = [l_{11}, l_{12}, l_{13}]^T \tag{4.4}$$

Using the above equation (4.4), the distance between the point and its corresponding epipolar line can be written as

$$\begin{aligned} d^2(\tilde{x}_2^j, F\tilde{x}_1^j) &= \frac{(\tilde{x}_2^{jT} F\tilde{x}_1^j)^2}{l_{21}^2 + l_{22}^2} \\ d^2(\tilde{x}_1^j, F^T\tilde{x}_2^j) &= \frac{(\tilde{x}_1^{jT} F^T\tilde{x}_2^j)^2}{l_{11}^2 + l_{12}^2}. \\ \text{Also} \quad \tilde{x}_2^{jT} F\tilde{x}_1^j &= \tilde{x}_1^{jT} F^T\tilde{x}_2^j. \end{aligned}$$

The distance measure from equation ((4.3)) can now be rewritten as

$$d^j = \left(\frac{1}{l_{11}^2 + l_{12}^2} + \frac{1}{l_{21}^2 + l_{22}^2}\right)\left((\tilde{x}_2^{jT} F\tilde{x}_1^j)^2\right). \tag{4.5}$$

Inliers are matches for which the condition $d^j < 1.0$ holds true. This condition implies that the obtained match is correct and the matches lie on their corresponding epipolar lines. The process of computing the fundamental matrix F using this method is described in Algorithm (7).

The feature descriptor FD_F describes the fundamental matrix and also defines a list of valid matches. The description is given below:

1. Fundamental Matrix F, a 3×3 matrix
2. List of Inliers L_i.

4.3 Model Construction

4.3.1 Data Representation

The data obtained during the data extraction process are represented by using a graph. The data includes a list of features in each image, the feature matches and information about the epipolar geometry for each image pair. The main purpose of this step is to represent the image

Algorithm 7 Robust Computation of Fundamental Matrix

INPUT: Set of initial point correspondences expressed in pixel coordinates $(x_1'^j, x_2'^j)$, $j = 1, 2, \ldots, n$,$n \geq 8$, Threshold τ, No. of iteration N(default 1000).

Match List L = List of all matches from the feature matching process L_m.
for $k \leftarrow 1, N$ **do**
    Select 8 random matches L_k from the match list L.
    Estimate F_k using the Eight Point algorithm, Algorithm (3).
    Set Number of Inliers $NL_k = 0$.
    for each match L_j in L_m **do**
        Compute d_j using equation 4.5.
        if $d_j < \tau$ **then**
            Increment NL_k.
        end if
        The matches that satisfy the condition is the $consensussetC_k$.
        Set $NL_k = Count(C_k)$.
    end for
end for
Choose F_k,C_k which has the maximum number of inliers, i.e. $Max(NL_k)$.

OUTPUT: Fundamental Matrix F, List of Inliers L_i.

collection and the data obtained from the images efficiently to facilitate easy access for later analysis. We use some definitions from the graph theory here.

Undirected Graph : An Undirected Graph is a graph whose edges are unordered pairs of vertices, i.e. each edge connects two vertices.

Complete Graph : A Complete Graph is a graph with an edge between every pair of vertices. A complete graph has $n(n-1)/2$ edges, where n is the number of vertices.

Null Graph : A Null Graph is a graph whose edge set is empty.

4.3.1.1 Initial Model

This step represents the initial state of the system. It describes the state when the input images are available and no processing has been performed on the image. It describes basic information about the image like its identification and its dimensions. The initial state of the system is represented using a null graph. The vertices of the graph describe the images in the image collection. This information is updated after more information about the image characteristics are available after further analysis.

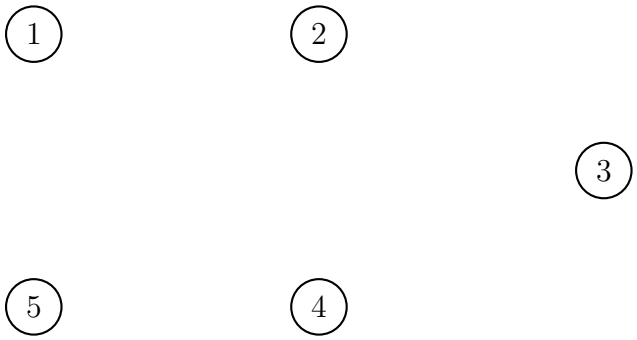


Figure 4.2: Initial Model

4.3.1.2 Refined Model

The null graph representation used to represent the initial model is refined to represent additional information. This is done once all the features from the image have been extracted and matched to determine the epipolar geometry. The initial model is augmented into a complete graph by adding an edge between every node in the null graph.

The nodes of the graph represent the images and any other data associated with that image. This data is available once the first step of the data extraction process is complete. The following are the data that are associated with an image and hence stored in the node of the graph.

1. Image identifier
2. Image dimensions
3. Transformation matrix

4. List of feature descriptor describing the corners.

The edges of the graph store information about the relationship between the images. The images in the collection are compared with each other to obtain the above data which are then stored in the edges. The information about the feature matches are obtained once the second step of data extraction is completed. These along with the information about the epipolar geometry computed from the matches are stored in the edges.

1. No. of feature matches.
2. List of feature descriptors describing the matches.
3. Feature descriptor for describing the fundamental matrix.
4. Feature descriptor for the epipole.

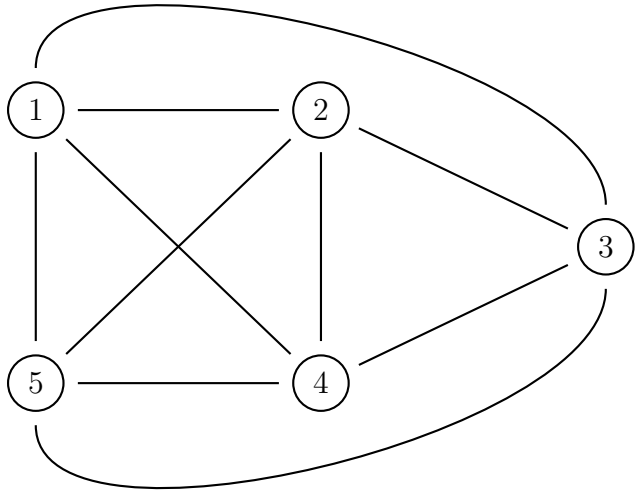


Figure 4.3: Graph based Data Representation

4.3.2 Analysis

The refined model is analyzed to estimate additional parameters and compute weighted measures based on these parameters. The measures obtained are weights that describe the various characteristics of the matches obtained. The aim is to determine the validity and match

strength between the images to generate a final model. This is done by traversing the graph and analyzing each edge and its corresponding vertices to infer statistical information from the store data. These are used to assert whether the images are connected or not connected. It also gives an indication about the strength of the connections in case the images are connected.
Computed Parameters

1. Correspondence information
    (a) Maximum possible matches (maxMatches)
    (b) No. of feature matches(numMatches)
2. NCC distribution
    (a) No. of matches with very high correlation (vhighNCC)
    (b) No. of matches with high correlation (highNCC)
    (c) No. of matches with average correlation (avgNCC)
3. Inlier information
    (a) No. of inliers (numInliers)

Correspondence information : The maximum possible feature matches is the minimum number of corner features between the concerned image pair.

$$maxMatches = Minimum(NumCorners(I_1), NumCorners(I_2))$$

where $NumCorners(I)$ denotes the number of corners features in image I.

The number of feature matches is given by the number of matches in the feature match list FD_{match}.

$$numMatches = Count(FD_{match})$$

NCC distribution : The NCC distribution gives an indication about the correlation characteristics of the selected feature matches. They are classified into matches with very high correlation, high correlation and average correlation. The threshold for these parameters

are given below:

$$\begin{aligned} vhighNCC &= Count(FD_{match}[i].NCCScore > \tau_{vhighNCC}) \\ highNCC &= Count(FD_{match}[i].NCCScore > \tau_{highNCC}) \\ avgNCC &= Count(FD_{match}[i].NCCScore > \tau_{avgNCC}) \end{aligned}$$

where $FD_{match}[i]$ denotes the feature descriptor for the i^{th} match and $\tau_{vhighNCC}, \tau_{highNCC}, \tau_{avgNCC}$ are the corresponding threshold values.

Inlier information The number of inliers is given by the total number of matches in the feature descriptor for inliers $FD_{inliers}$.

$$numInliers = Count(FD_{inliers})$$

4.3.2.1 Compatibility Measurements:

Compatibility Measurements are obtained from the computed parameters. They are described below:

Feature Ratio: This is the ratio of the number of features between the two images.

Match Percentage: This is the percentage of obtained matches. It is obtained as follows:

$$pctMatches = \left(\frac{numMatches}{maxMatches}\right) \times 100$$

Inlier Percentage: This is the percentage of inliers out of the obtained matches. The threshold for a match to be an inlier is 1pixel. It is given by the ratio between the number of inliers and the number of matches.

$$pctInliers = \left(\frac{numInliers}{numMatches}\right) \times 100$$

4.3.2.2 Compatibility Measures

The following are the weighted compatibility measures that are computed from the compatibility measurements and the computed parameters which are then used to determine the connection strength between the images.

1. $wtMatches(W_m)$ - Indicates percentage of correspondences,

2. $wtCorrelation(W_c)$ - Indicates correlation, and

3. $wtInliers(W_i)$ - Indicates percentage of inliers.

The weighted measures are scaled values and the lower and the upper values used for scaling each of these measures are given by $LValue$ and $UValue$ respectively.

The weighted measure for the percentage matches is a scaled value between 0 and 1. It is denoted by W_m.

$$W_m = \frac{numMatches - LValue}{UValue - LValue}$$

The weighed measure for indicating the correlation of the correspondences is obtained as a scaled value between 0 and 2, denoted by W_{c1}. The correspondences which have very high correlation values are given more weightage and is denoted by W_{c2}.

$$\begin{aligned} W_{c1} &= \frac{(highNCC + vhighNCC) - LValue}{UValue - LValue} \\ W_{c1} &= \frac{vhighNCC - LValue}{UValue - LValue} \\ W_c &= W_{c1} + W_{c2} \end{aligned}$$

The weighed measure for indicating the percentage of inliers is a scaled value between 0 and 2 and is denoted by W_i.

$$W_i = \frac{numInliers - LValue}{UValue - LValue}$$

The final weighted measure is obtained as a sum of all the measures obtained above.

$$W = W_m + W_c + W_i$$

4.3.3 Model Generation

This is the last stage of the system and is concerned with generating a model to represent the connectivity information between the images. The results obtained from analysis is used to this model. The model used here is an weighted, undirected graph. The reason for using a

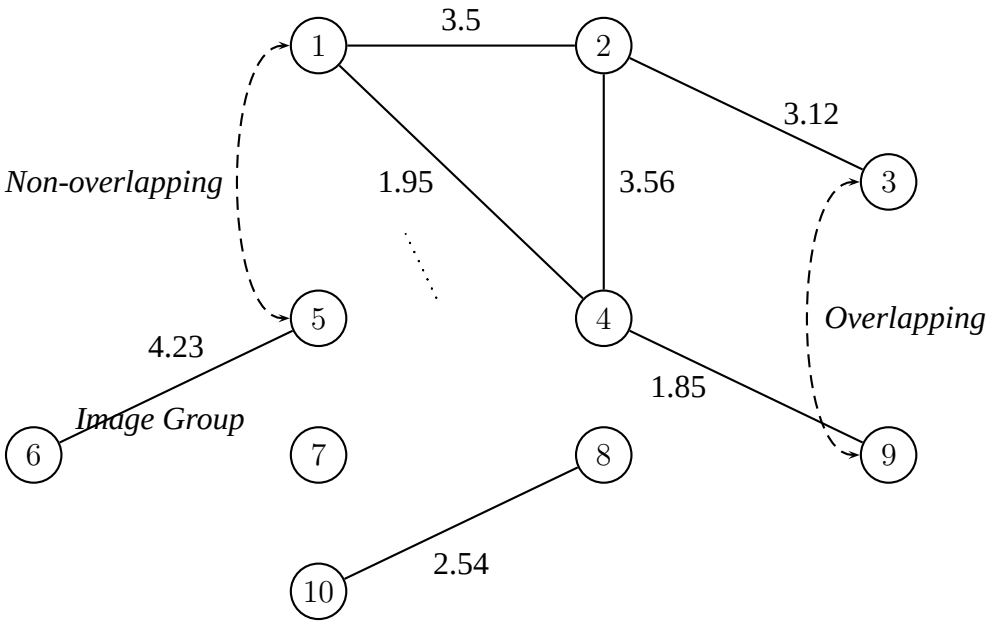


Figure 4.4: Final Model

graph model is that it is a simple and effective method for storing connectivity information and grouping them into distinct scenes. A measure of the connectivity strength is also determined and stored in the model.

The model is generated by creating nodes representing all the images in the collection. The results obtained from each image pair are then used to determine the connectivity between them by checking the weighted quantities. An image pair is considered to be connected if the total weighted quantity(W) has a positive value. The strength of the matches is described by this value which is a positive real value (0 - 5) if the images match.

The overlapping images are denoted by the presence of an edge between the nodes which represents the images. It basically indicates that the two images match. The image groups are represented by the subgraphs in the graph. An image group comprises of overlapping images and other images that are part of the same scene. All the images in the group may not have an edge between them but they are connected with each other through other images or views of the scene. Non-overlapping images do not belong to the same group. The nodes representing the images are in different subgraphs.

CHAPTER V
RESULTS

This section details the results obtained from analyzing a sample comprising of 37 images. The obtained results are broken down into four sections. The first section is a list of features that were obtained from the feature extraction stage. The second is obtained from matching the features extracted from each image. It is a set of matching feature for each image pair in the collection and their corresponding fundamental matrix. The third group contains information obtained by analyzing the data from feature extraction and matching. The last group consists of a model which represents the final results. It provides a classification of the input images into distinct groups.

The results are presented here using a grayscale image matrix. The darker sections of the image indicates favorable results while lighter sections imply less favorable results. The rows of the image matrix represents the results of comparing an image with all the other images. The top left corner is the comparison of the first image with itself, i.e. image id (0) and increases sequentially till 36 in both the x and y directions. The main diagonal of the image matrix is dark since it represents the comparison of the image with itself.

The program was tested on a Pentium 4 machine with a clock speed of 2.4GHz and 1GB RAM. The Operating System used was Gentoo v2.4, a linux distribution. The program was completely written in C++ and the GUI interface is based on QT Toolkit. A collection of 37 images consisting of different image groups were used to obtain the results. The sample images used are displayed in Figure 5.1.

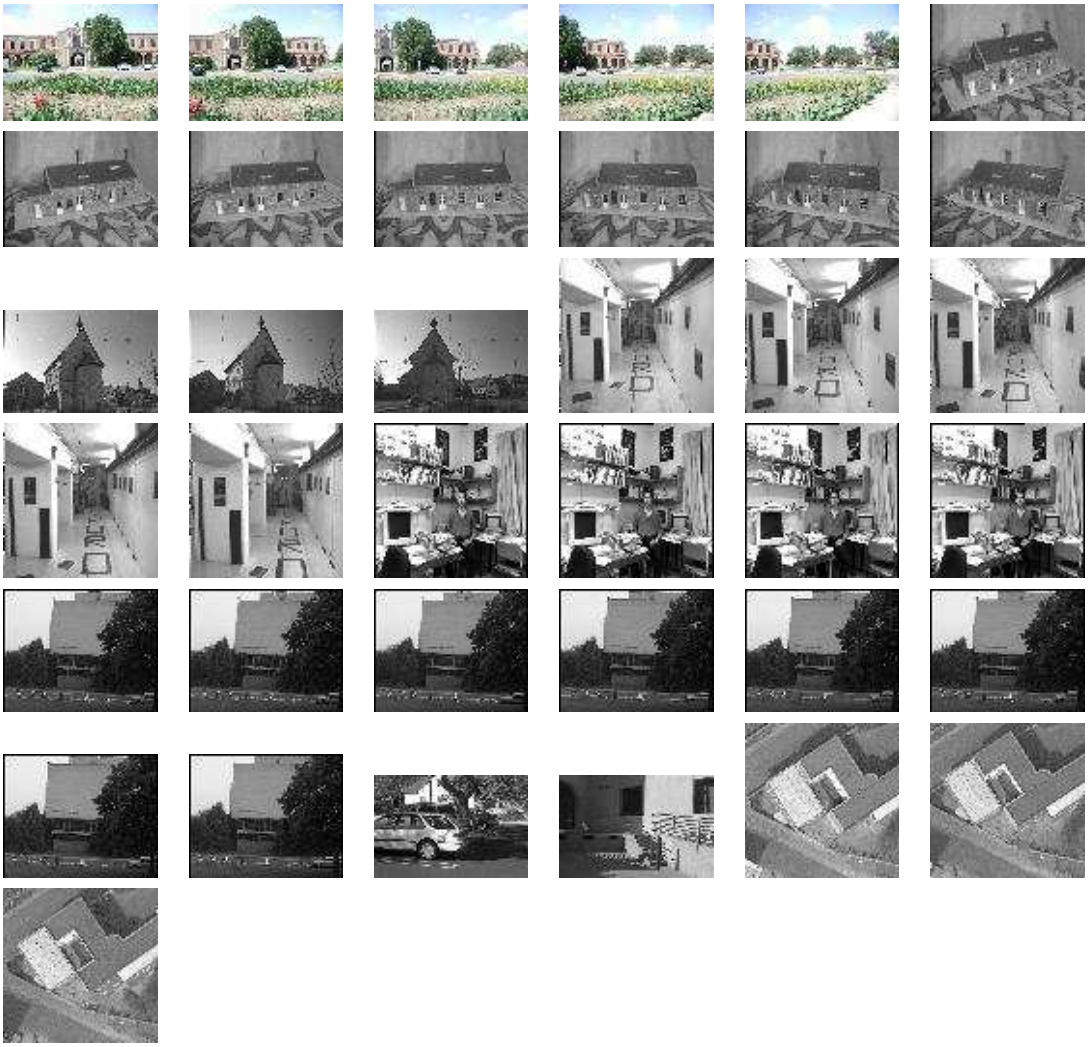


Figure 5.1: The images used in the study.

5.1 Features

The Corner Detector was applied to each image. It took approximately 2.2 seconds for each image. The average number of corners per image was 75. The features are displayed in Figure 5.2 as a bar graph with Image ID in the x-axis and the number of features in the y-axis. An example is given in Figure 5.3.

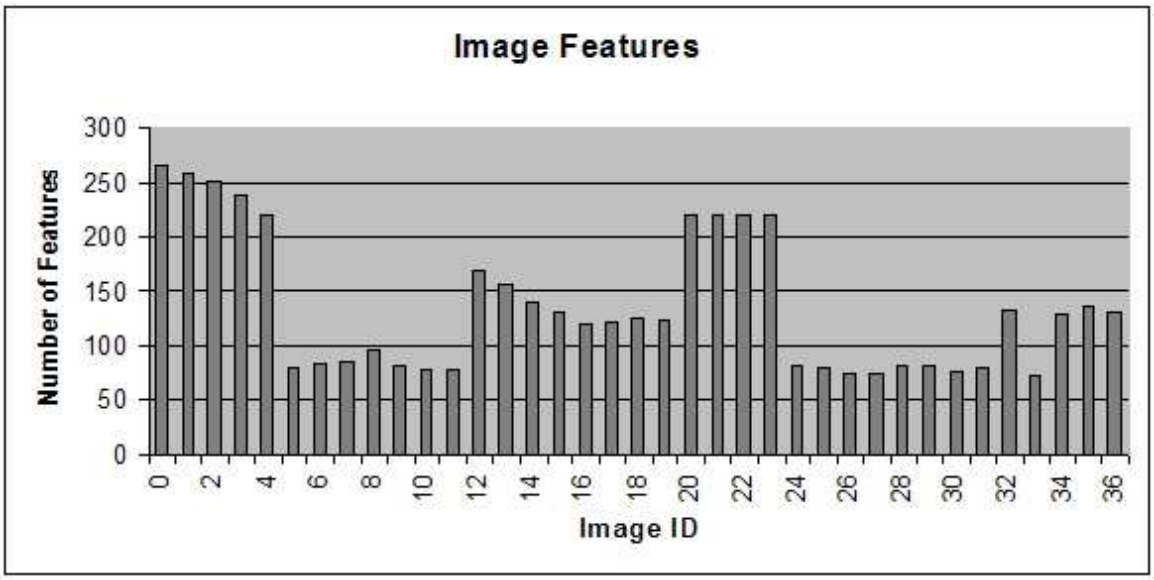


Figure 5.2: Number of Features per image
(Radius:6,Threshold:4.5,Boundary:10, Sigma:1)

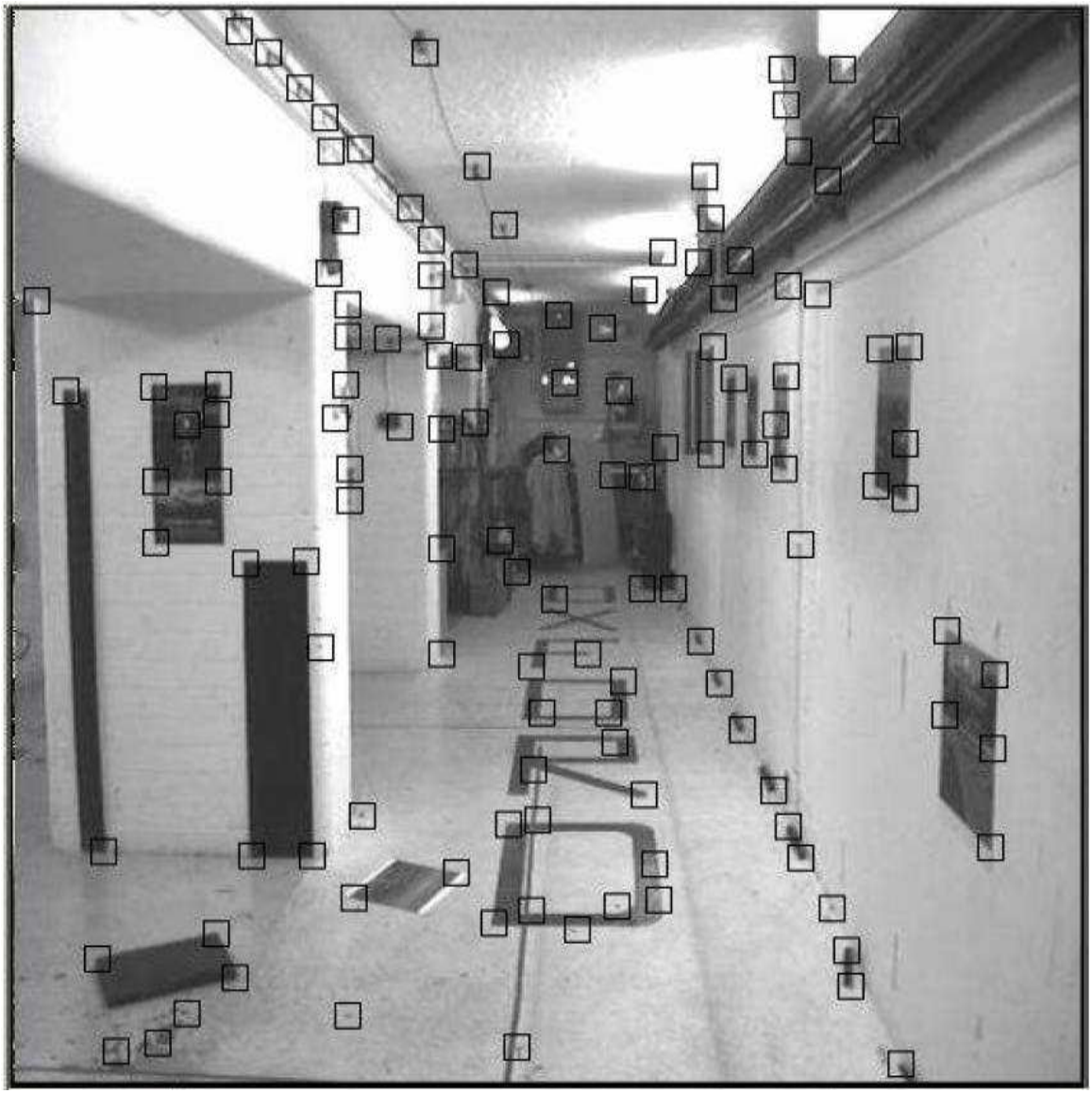


Figure 5.3: Image Features

5.2 Matches and Fundamental Matrix

The matches are obtained by comparing the features between the two images. This is a time consuming process as it involves matching each feature in the firs image with all other features in the second image. The features which has the highest NCC score is taken into account. The following information is obtained at the end of this matching process. The total number of image comparisons that were made is 666 and the time for each comparison was approximately 0.6 seconds. An example feature match is given in Figure 5.4.

1. Selected Matches (results are given in Table 1 in the appendix)

2. Match Distribution - Groups the matches into categories (Very High, High, Average) based on NCC Score.(results are given in Tables 2,3 and 4 in the appendix)

The fundamental matrix F is calculated from the list of matches. It is achieved by choosing 8 matches in random and calculating F from it using the 8 Point Algorithm. The process is repeated over for a couple thousand times and the best set which has the largest number of inliers is chosen as F. The result is a 3×3 matrix.

An inlier is a match that is very close to the actual match. It is determined by finding the distance between the feature location and epipolar line. The distance should be less than a pixel in order for a match to be considered as an inlier. The number of inliers is determined checking all the matches.(results are given in Table 5 in the appendix)

Figure 5.4: Image Match

5.3 Analysis

This section contains a list of compatibility measurement for each image pair in the collection. These are then weighed appropriately to arrive at a final compatibility measure for the image pair. This is then used to determine whether the images are connected. The following are the compatibility measurements that are obtained for each image pair.

Image Pair Compatibility Measurements:

1. Percentage of Matches - It is the percentage of selected matches out of the total matches.
2. Feature Ratio - It is the ratio between the features in the two images.
3. Percentage of Inliers - It is the percentage of inliers out the the selected matches.

The above measurements are then weighed appropriately to obtain compatibility measures. This is done by applying a suitable scale to each of the calculated values and estimating their contribution to the total value. A negative weightage is given to those values if it falls below the preset limit. A final compatibility measure is then obtained which indicates the strength of the connection between the image pair. A very low or negative value indicates that the images are not connected. The results are represented using a grayscale image with darker shades indicating maximum compatibility while lighter shades indicates minimum compatibility. The results are shown in Figures 5.5 - 5.11.

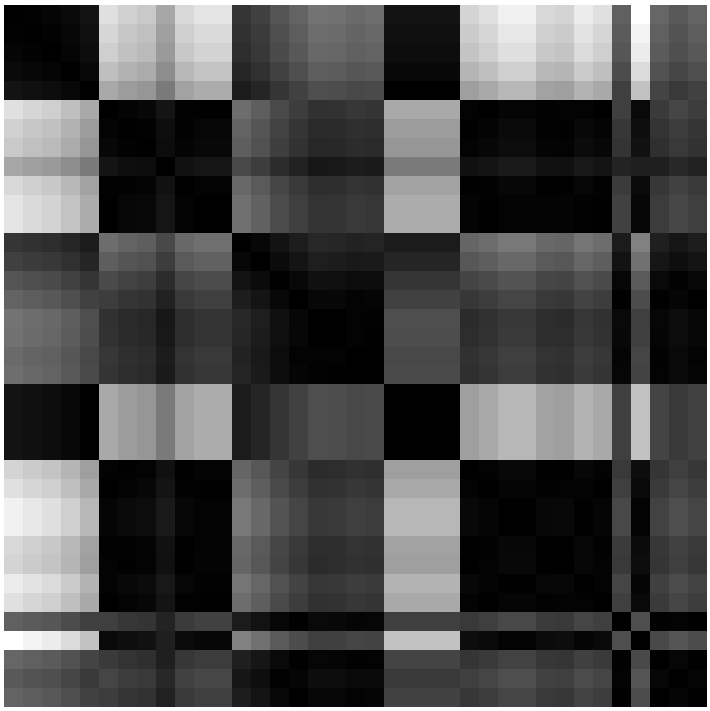


Figure 5.5: Compatibility Measurements – Feature Ratio

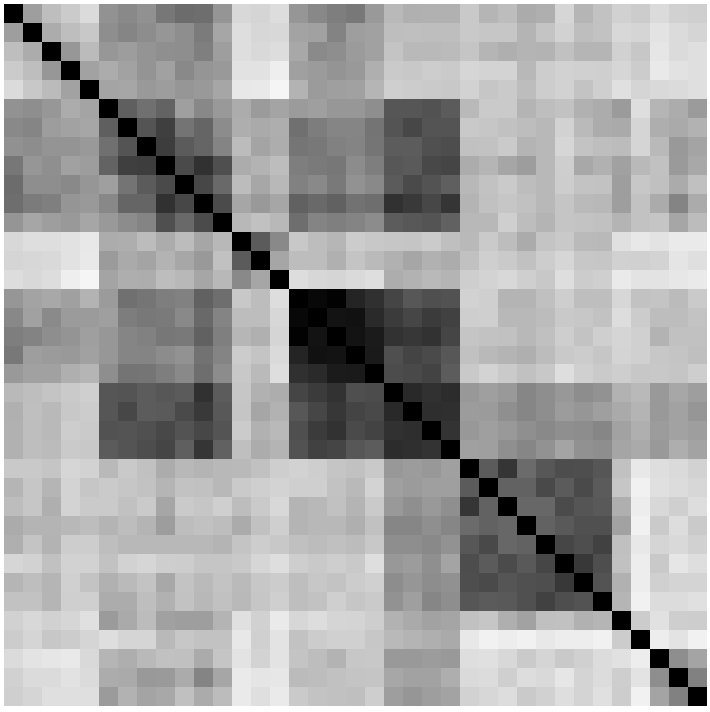


Figure 5.6: Compatibility Measurements – Match Percentage

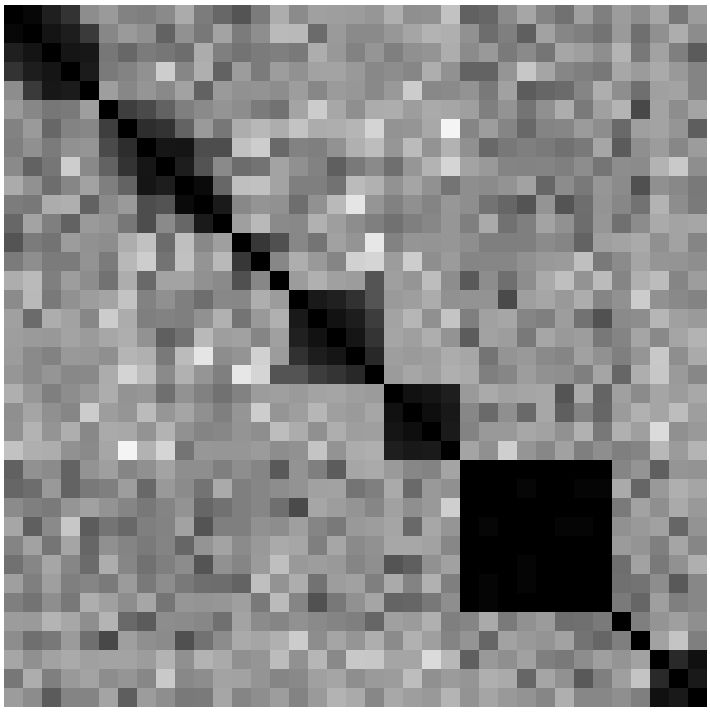


Figure 5.7: Compatibility Measurements – Inlier Percentage

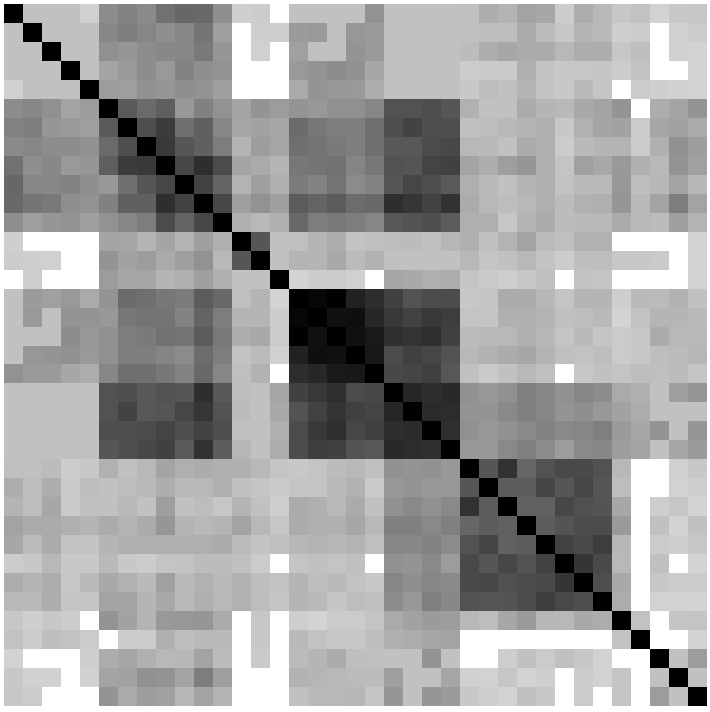


Figure 5.8: Compatibility Measures – Matches

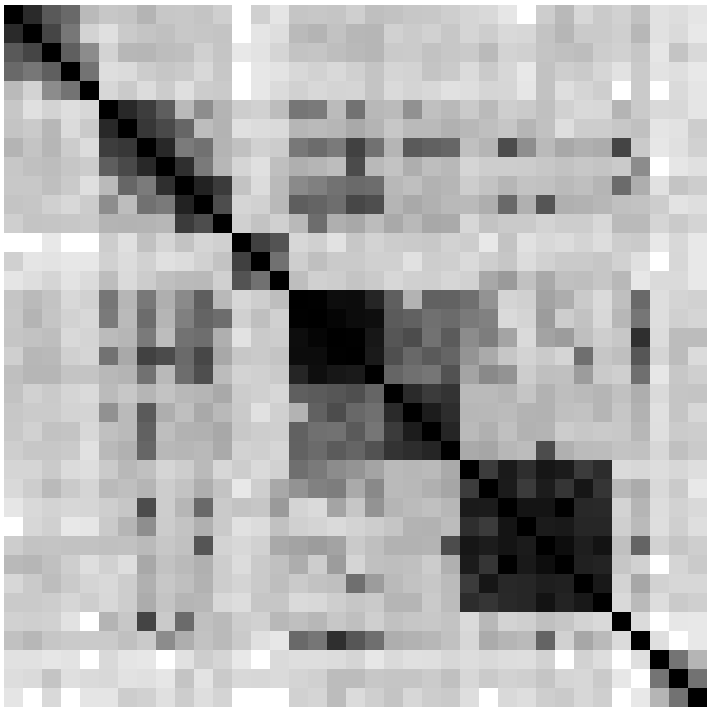


Figure 5.9: Compatibility Measures – Correlation

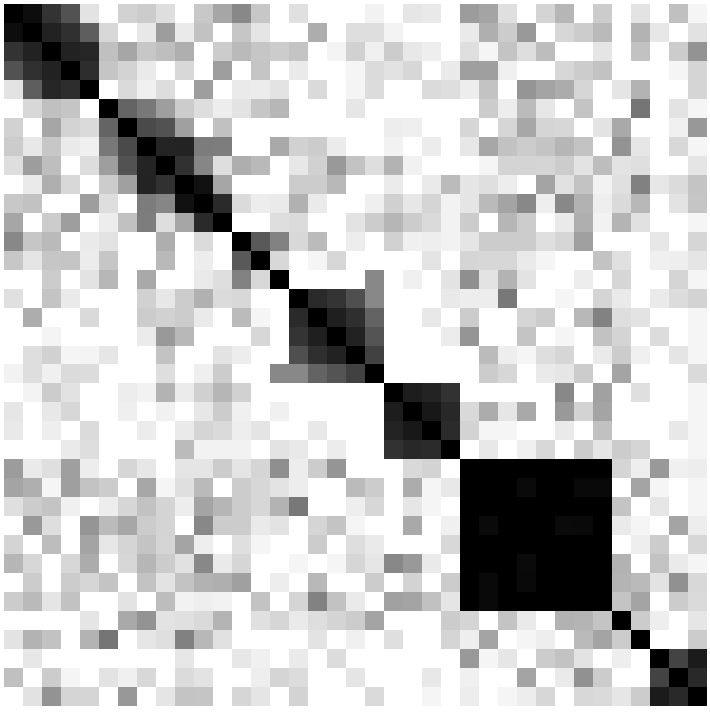


Figure 5.10: Compatibility Measures – Inliers

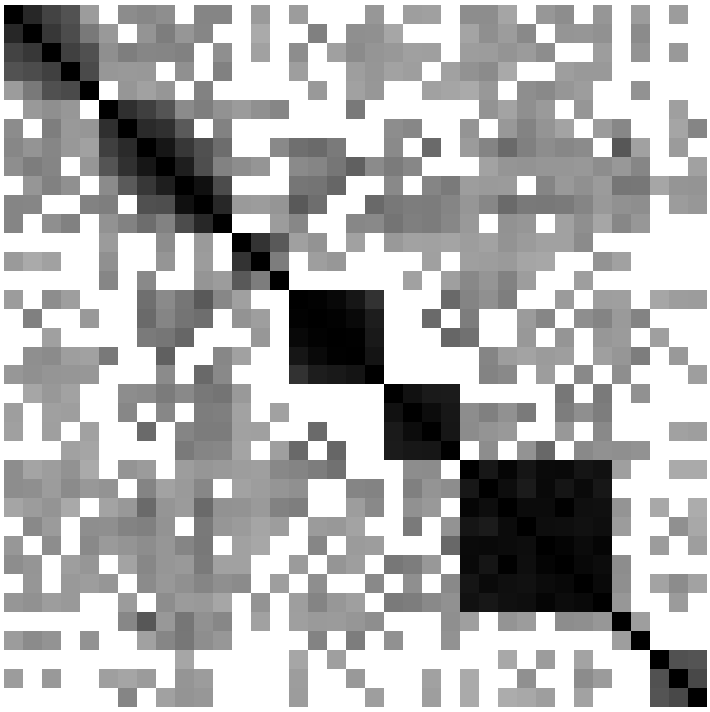


Figure 5.11: Compatibility Measures – Total Weight

5.4 Final Model

This section describes the final result. It specifies that total number of groups in the given image collection and identifies which image belongs to which group. It also gives the total number of images that are connected. The results are derived by constructing a group using images as nodes and using the weighted values to determine the image connections. The results for each group can be found in Figures 5.12 - 5.20.

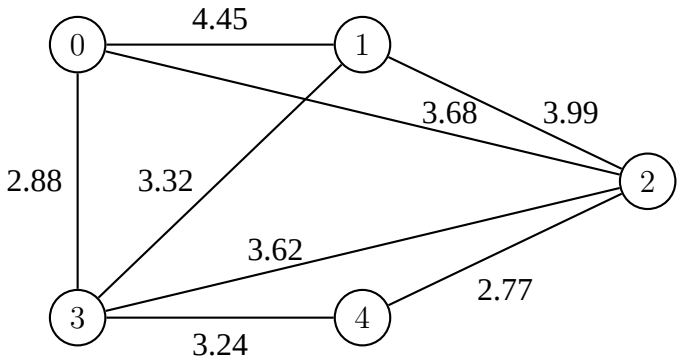


Figure 5.12: Group 1

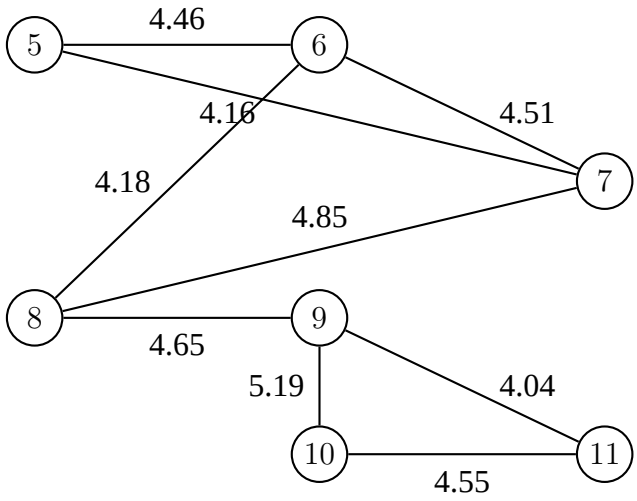


Figure 5.13: Group 2

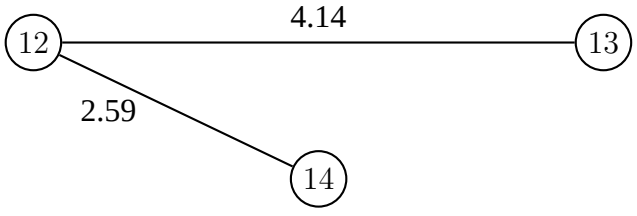


Figure 5.14: Group 3

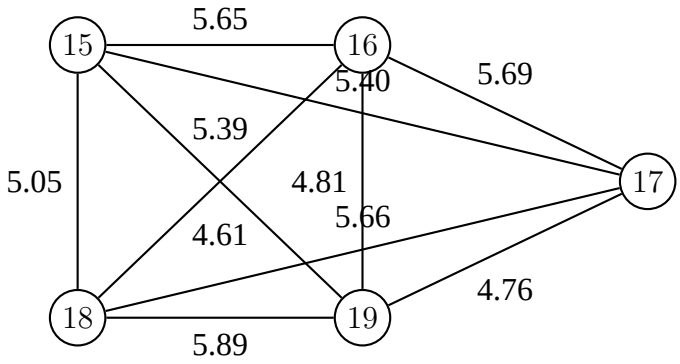


Figure 5.15: Group 4

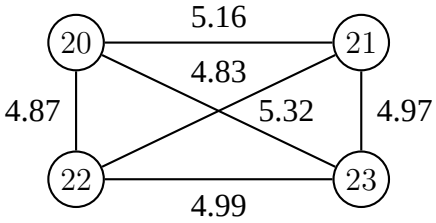
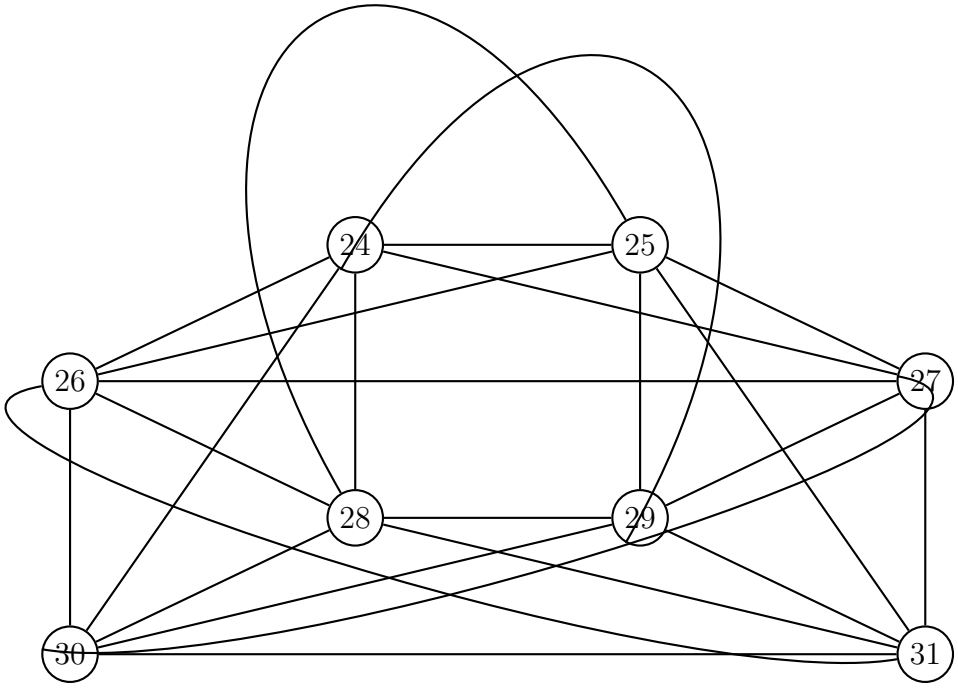


Figure 5.16: Group 5



(24 , 25)	4.964617	(25 , 26)	5.284704	(28 , 29)	5.588207
(24 , 26)	5.540559	(25 , 27)	4.840260	(28 , 30)	5.359347
(24 , 27)	5.031061	(25 , 28)	5.413778	(28 , 31)	5.547244
(24 , 28)	5.398527	(25 , 29)	5.044327		
(24 , 29)	5.402478	(25 , 30)	5.386078	(29 , 30)	5.453460
(24 , 30)	5.047290	(25 , 31)	4.957371	(29 , 31)	5.306954
(24 , 31)	5.160620				
		(27 , 28)	5.284704	(30 , 31)	5.404958
(26 , 27)	5.339246	(27 , 29)	5.262121		
(26 , 28)	5.339246	(27 , 30)	5.173254		
(26 , 29)	5.686396	(27 , 31)	5.260211		
(26 , 30)	5.223569				
(26 , 31)	5.186364				

Figure 5.17: Group 6

Figure 5.18: Group 7

Figure 5.19: Group 8

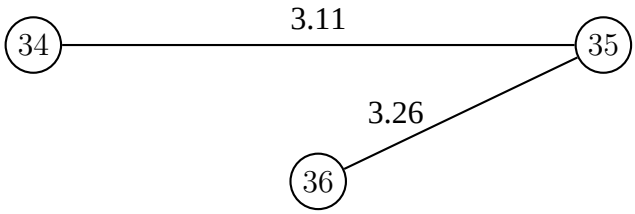


Figure 5.20: Group 9

CHAPTER VI
CONCLUSION

The main aim of this work was to analyze a collection of images and generate a graph model to determine the connectivity between the images and classify them into distinct scenes. A large number of images from different scenes were tested and the model generated was accurate in most cases. The main application of this work is in automating the process of classifying images and grouping them into scenes which is usually the first and an important step in many vision based applications.

The classification of the input images into their groups was mostly successful provided the input images match the specified requirements. The error case that was generally noticed was the failure to group images that are part of the same scene, i.e. false negatives. The other case of matching two incorrect images is rare and was not noticed in the test images used, i.e. no false positives. Almost all the cases of false negatives were due to the wrong matches obtained during the feature matching process. This underlines the fact that the accuracy of the results is highly dependent on the initial set of feature matches extracted from the image. This effect is particularly visible in the case of large baseline where is a significant difference between the two views. The other degenerate configuration is a simple translation motion between images which has similar features repeated across the image. A classic example is image of a wall containing multiple windows which are very similar in their appearance.

Future work in this direction involves speeding up the classification process using heuristics to direct the matching process. The other important addition is to determine the camera motion between the images. This would make the results even more useful in many practical applications like generating a image based virtual tour sequence from just a collection of images or in image based modelling and rendering.

BIBLIOGRAPHY

B.D.Lucas, T.Kanade, 1981. An iterative image registration technique with an application to stereo vision. In: International Joint Conference on Artificial Intelligence. pp. 674–679.

Carson, C., Thomas, M., Belongie, S., Hellerstein, J. M., Malik, J., 1999. Blobworld: A system for region-based image indexing and retrieval. In: Third International Conference on Visual Information Systems. Springer.

David A. Forsyth, J. P., 2003. Computer Vision A Modern Approach. Pearson Education.

Debevec, P. E., 1996. Modeling and rendering architecture from photographs. Ph.D. thesis, University of California, Berkeley.

Debevec, P. E., Taylor, C. J., Malik, M., 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In: Proceedings of the ACM Conference on Computer Graphics. ACM, New York, pp. 11–20.

Deriche, R., Giraudon, G., 1993. A computational approach for corner and vertex detection. IJCV 10 (2), 101–124.

Emanuele Trucco, A. V., 1998. Introductory Techniques for 3-D Computer Vision. Prentice Hall.

Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., Equitz, W., 1994. Efficient and effective querying by image content. Journal of Intelligent Information Systems 3 (3/4), 231–262.

Harris, C., Stephens, M., 1988. A combined corner and edge detector. In: In Proceedings of the Alvey Conference. pp. 189–192.

H.C.Longuet-Higgins, 1981. A computer algorithm for reconstructing a scene from two projections. Nature 293 (10), 133–135.

J.Canny, 1986. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8, 679–698.

Jethwa, M., Antone, M., Bosse, M., Master, N., Coorg, S., Teller, S., Bodnar, Z., 2003. Calibrated, registered images of an extended urban area.
URL `http://citeseer.ist.psu.edu/565969.html`

Laaksonen, M. K. A., Oja, E., Jun. 2001. Comparison of techniques for content-based image retrieval. In: Proceedings of SCIA 2001. Bergen,Norway.

Lewis, J., 1995. Fast normalized cross-correlation.

Luong, Q., Faugeras, O., 1996. The fundamental matrix: Theory, algorithms, and stability analysis. International Journal of Computer Vision 17, 43–75.

Ma, W.-Y., Manjunath, B. S., 1999. Netra: A toolbox for navigating large image databases. Multimedia Systems 7 (3), 184–198.

Ma, Y., Soatto, S., Kosecka, J., Sastry, S. S., 2004. An Invitation to 3-D Vision. Springer.

Minimal, M. W., 2004. Nonlinear estimation of the fundamental.
URL citeseer.ist.psu.edu/700675.html

Newsam, M. S., Sumengen, B., Manjunath, B. S., Sep. 2001. Category-based image retrieval. In: IEEE International Conference on Image Processing (ICIP 2001), Special Session on Multimedia Indexing, Browsing and Retrieval. Thessalonica, Greece.

R.I.Hartley, 1995. In defense of the 8-point algorithm. In: 5th International Conference on Computer Vision. pp. 1064–1070.

Smith, J. R., Chang, S.-F., 1996. Visualseek: A fully automated content-based image query system. In: ACM Multimedia. pp. 87–98.

Teller, S., 1998. Toward urban model acquisition from geo-located images. In: Werner, B. (Ed.), Proceedings of the Conference on Computer Graphics and Applications (Pacific-Graphics'98). IEEE, Los Alamitos, pp. 45–52.

Tomasi, C., Kanade, T., April 1991. Detection and tracking of point features. Tech. Rep. CMU-CS-91-132, Carnegie Mellon University.

Tomasi, C., Kanade, T., March 1992. Shape and Motion from Image Streams: a Factorization Method, Full Report on the Orthographic Case. Tech. Rep. CMU-CS-92-104.

Zhang, Z., 1996. Determining the epipolar geometry and its uncertainty: A review. Tech. Rep. 2927, Sophia-Antipolis Cedex, France.

Table 1: Feature Matches(Radius:6,Threshold:0.69)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	-	99	73	60	45	33	37	36	48	43	41	32	36	35	28	53	54	58	62	49	67	73	72	72	21	25	22	26	27	18	24	22	31	18	26	32	31
1	99	-	86	69	57	34	38	37	39	35	38	28	33	34	30	48	45	56	48	46	64	63	64	64	21	21	20	24	22	19	22	22	28	16	23	28	29
2	73	86	-	97	66	32	32	35	41	35	37	30	33	33	28	46	53	52	49	46	60	67	67	66	22	25	25	24	26	23	24	25	31	19	22	28	25
3	60	69	97	-	82	27	31	35	36	36	32	31	31	29	21	49	47	50	50	41	56	57	54	56	18	18	17	23	20	19	18	19	29	18	21	25	25
4	45	57	66	82	-	29	30	34	37	33	31	28	28	26	18	42	47	46	47	39	53	55	56	57	19	21	19	22	20	19	21	19	25	16	27	28	25
5	33	34	32	27	29	-	45	41	44	30	35	31	27	31	29	31	30	32	32	34	47	47	48	46	24	20	19	24	23	22	25	27	31	14	25	27	31
6	37	38	32	31	30	45	-	50	55	41	43	34	28	29	28	43	40	38	38	42	50	53	50	50	21	21	21	22	25	19	23	27	29	16	28	29	27
7	36	37	35	35	34	41	50	-	57	47	43	36	25	31	29	43	42	40	39	43	49	49	52	53	23	22	19	24	25	18	21	23	25	16	26	34	31
8	48	39	41	36	37	44	55	57	-	51	55	48	33	31	28	46	47	48	42	46	57	56	61	62	28	25	27	29	25	20	27	26	36	22	27	38	34
9	43	35	35	36	33	30	41	47	51	-	50	42	24	29	25	38	36	40	34	36	47	51	49	46	25	22	19	22	25	20	21	22	31	19	23	28	23
10	41	38	37	32	31	35	43	43	55	50	-	49	28	31	28	44	42	44	40	41	55	53	50	54	25	22	20	21	24	21	23	24	30	20	27	36	27
11	32	28	30	31	28	31	34	36	48	42	49	-	24	22	24	41	36	37	36	34	45	46	47	44	24	24	18	22	25	21	21	22	25	20	21	28	23
12	36	33	33	31	28	27	28	25	33	24	28	24	-	92	63	34	35	38	31	32	45	46	43	49	25	20	22	26	22	21	23	24	25	12	23	22	21
13	35	34	33	29	26	31	29	31	31	29	31	22	92	-	51	38	35	40	34	37	52	57	52	54	23	19	19	21	20	18	20	21	31	17	29	24	25
14	28	30	28	21	18	29	28	29	28	25	28	24	63	51	-	26	26	27	26	24	48	46	43	45	20	18	16	18	21	16	19	19	21	13	23	23	21
15	53	48	46	49	42	31	43	43	46	38	44	41	34	38	26	-	99	102	93	87	84	77	80	79	19	19	24	24	23	20	22	23	28	20	35	39	33
16	54	45	53	47	47	30	40	42	47	36	42	36	35	35	26	99	-	95	96	87	81	77	81	79	20	19	23	23	24	21	20	22	24	18	35	35	33
17	58	56	52	50	46	32	38	40	48	40	44	37	38	40	27	102	95	-	96	93	85	83	85	78	22	22	21	23	23	20	21	19	27	17	39	34	34
18	62	48	49	50	47	32	38	39	42	34	40	36	31	34	26	93	96	96	-	95	77	81	79	74	23	24	24	25	24	21	19	21	28	17	33	34	36

Table 1 – Continued

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
19	49	46	46	41	39	34	42	43	46	36	41	34	32	37	24	87	87	93	95	-	77	78	80	71	21	18	19	21	21	16	18	20	30	19	32	33	30
20	67	64	60	56	53	47	50	49	57	47	55	45	45	52	48	84	81	85	77	77	-	160	155	158	33	30	30	34	35	33	33	33	43	22	51	49	50
21	73	63	67	57	55	47	53	49	56	51	53	46	46	57	46	77	77	83	81	78	160	-	157	156	32	31	32	34	35	32	31	30	46	23	51	51	53
22	72	64	67	54	56	48	50	52	61	49	50	47	43	52	43	80	81	85	79	80	155	157	-	154	33	30	30	32	36	35	33	36	49	24	50	51	50
23	72	64	66	56	57	46	50	53	62	46	54	44	49	54	45	79	79	78	74	71	158	156	154	-	31	30	32	34	34	30	32	34	48	25	51	46	48
24	21	21	22	18	19	24	21	23	28	25	25	24	25	23	20	19	20	22	23	21	33	32	33	31	-	46	52	40	48	51	47	47	23	12	16	17	19
25	25	21	25	18	21	20	21	22	25	22	22	24	20	19	18	19	19	22	24	18	30	31	30	30	46	-	42	42	50	46	48	46	21	10	15	16	17
26	22	20	25	17	19	19	21	19	27	19	20	18	22	19	16	24	23	21	24	19	30	32	30	32	52	42	-	41	41	47	45	44	25	11	16	18	15
27	26	24	24	23	22	24	22	24	29	22	21	22	26	21	18	24	23	23	25	21	34	34	32	34	40	42	41	-	42	44	46	46	28	10	19	15	19
28	27	22	26	20	20	23	25	25	25	25	24	25	22	20	21	23	24	23	24	21	35	35	36	34	48	50	41	42	-	52	47	52	23	12	18	17	19
29	18	19	23	19	19	22	19	18	20	20	21	21	21	18	16	20	21	20	21	16	33	32	35	30	51	46	47	44	52	-	52	49	23	11	21	14	16
30	24	22	24	18	21	25	23	21	27	21	23	21	23	20	19	22	20	21	19	18	33	31	33	32	47	48	45	46	47	52	-	46	25	11	18	17	17
31	22	22	25	19	19	27	27	23	26	22	24	22	24	21	19	23	22	19	21	20	33	30	36	34	47	46	44	46	52	49	46	-	25	10	16	16	15
32	31	28	31	29	25	31	29	25	36	31	30	25	25	31	21	28	24	27	28	30	43	46	49	48	23	21	25	28	23	23	25	25	-	18	24	32	32
33	18	16	19	18	16	14	16	16	22	19	20	20	12	17	13	20	18	17	17	19	22	23	24	25	12	10	11	10	12	11	11	10	18	-	11	13	10
34	26	23	22	21	27	25	28	26	27	23	27	21	23	29	23	35	35	39	33	32	51	51	50	51	16	15	16	19	18	21	18	16	24	11	-	55	47
35	32	28	28	25	28	27	29	34	38	28	36	28	22	24	23	39	35	34	34	33	49	51	51	46	17	16	18	15	17	14	17	16	32	13	55	-	61
36	31	29	25	25	25	31	27	31	34	23	27	23	21	25	21	33	33	34	36	30	50	53	50	48	19	17	15	19	19	16	17	15	32	10	47	61	-

Table 2: Feature Matches with very high correspondence

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	-	57	33	26	10	2	2	4	2	2	5	2	0	0	2	9	7	9	8	10	7	6	7	7	3	1	2	1	2	2	1	3	4	3	0	0	0
1	57	-	42	30	12	1	1	4	4	4	3	2	0	0	2	7	7	11	8	7	3	5	5	5	3	2	1	3	2	1	2	2	3	1	0	0	0
2	33	42	-	33	21	2	2	7	4	3	3	2	1	1	2	8	9	9	7	9	6	7	9	7	1	0	1	2	2	1	1	1	3	3	0	0	0
3	26	30	33	-	31	3	2	5	5	3	5	3	1	1	0	5	6	5	6	6	5	5	5	5	2	1	2	2	2	2	3	2	3	2	0	0	0
4	10	12	21	31	-	2	1	1	1	1	2	2	0	0	1	5	6	7	5	4	5	7	6	6	1	1	1	1	1	1	1	1	1	1	0	1	0
5	2	1	2	3	2	-	20	19	16	7	12	7	1	3	5	9	9	7	8	8	11	12	8	11	2	2	2	1	2	2	1	2	6	1	0	1	1
6	2	1	2	2	1	20	-	27	22	14	9	5	0	1	5	7	6	6	6	8	8	10	6	7	2	1	2	0	1	2	1	2	6	0	1	1	2
7	4	4	7	5	1	19	27	-	36	18	16	7	0	2	4	11	14	14	14	12	12	13	14	14	2	3	5	6	4	3	3	4	9	3	1	0	0
8	2	4	4	5	1	16	22	36	-	27	20	7	1	2	3	11	11	11	12	11	12	12	13	14	2	2	2	4	3	2	3	2	8	6	0	0	0
9	2	4	3	3	1	7	14	18	27	-	24	14	1	2	3	11	10	11	9	11	10	10	8	10	3	2	2	2	2	3	2	2	8	2	0	1	0
10	5	3	3	5	2	12	9	16	20	24	-	19	2	2	3	15	13	13	16	17	10	9	10	12	2	5	5	5	6	5	5	3	6	2	2	1	1
11	2	2	2	3	2	7	5	7	7	14	19	-	3	1	3	6	10	8	8	8	7	6	8	8	3	2	3	2	2	3	2	2	3	1	0	0	0
12	0	0	1	1	0	1	0	0	1	1	2	3	-	55	25	1	2	1	1	0	1	1	1	3	0	0	0	0	0	0	0	0	1	1	1	2	0
13	0	0	1	1	0	3	1	2	2	2	2	1	55	-	16	3	2	2	1	1	2	3	3	5	0	0	0	1	0	0	0	0	1	0	1	1	2
14	2	2	2	0	1	5	5	4	3	3	3	3	25	16	-	5	6	6	6	5	4	4	4	4	3	2	3	3	3	2	3	3	2	0	0	0	0
15	9	7	8	5	5	9	7	11	11	11	15	6	1	3	5	-	84	79	69	62	22	17	22	23	7	7	6	5	7	5	4	4	5	8	2	2	1
16	7	7	9	6	6	9	6	14	11	10	13	10	2	2	6	84	-	75	78	66	25	22	25	24	6	6	5	4	6	2	3	4	5	5	1	1	1
17	9	11	9	5	7	7	6	14	11	11	13	8	1	2	6	79	75	-	83	66	27	22	27	22	7	8	6	5	7	6	4	4	5	6	1	0	1
18	8	8	7	6	5	8	6	14	12	9	16	8	1	1	6	69	78	83	-	74	26	22	24	25	6	7	7	6	5	3	5	4	4	7	1	0	1

Table 2 – Continued

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
19	10	7	9	6	4	8	8	12	11	11	17	8	0	1	5	62	66	66	74	-	22	21	24	24	6	6	6	5	4	3	5	4	5	6	1	2	0
20	7	3	6	5	5	11	8	12	12	10	10	7	1	2	4	22	25	27	26	22	-	99	95	85	6	6	5	5	5	4	4	5	6	4	1	2	1
21	6	5	7	5	7	12	10	13	12	10	9	6	1	3	4	17	22	22	22	21	99	-	105	95	7	6	6	6	6	5	5	6	6	3	2	1	0
22	7	5	9	5	6	8	6	14	13	8	10	8	1	3	4	22	25	27	24	24	95	105	-	100	6	4	6	4	4	4	5	6	6	2	1	1	0
23	7	5	7	5	6	11	7	14	14	10	12	8	3	5	4	23	24	22	25	24	85	95	100	-	7	4	6	6	9	5	5	6	7	2	2	1	1
24	3	3	1	2	1	2	2	2	2	3	2	3	0	0	3	7	6	7	6	6	6	7	6	7	-	25	31	23	35	31	30	30	2	2	0	0	0
25	1	2	0	1	1	2	1	3	2	2	5	2	0	0	2	7	6	8	7	6	6	6	4	4	25	-	22	24	31	25	31	26	2	1	2	0	1
26	2	1	1	2	1	2	2	5	2	2	5	3	0	0	3	6	5	6	7	6	5	6	6	6	31	22	-	28	28	35	25	27	1	1	1	0	1
27	1	3	2	2	1	1	0	6	4	2	5	2	0	1	3	5	4	5	6	5	5	6	4	6	23	24	28	-	30	25	29	32	1	1	1	0	1
28	2	2	2	2	1	2	1	4	3	2	6	2	0	0	3	7	6	7	5	4	5	6	4	9	35	31	28	30	-	39	32	35	3	3	1	0	1
29	2	1	1	2	1	2	2	3	2	3	5	3	0	0	2	5	2	6	3	3	4	5	4	5	31	25	35	25	39	-	31	30	3	1	1	0	1
30	1	2	1	3	1	1	1	3	3	2	5	2	0	0	3	4	3	4	5	5	4	5	5	5	30	31	25	29	32	31	-	31	2	2	1	0	1
31	3	2	1	2	1	2	2	4	2	2	3	2	0	0	3	4	4	4	4	4	5	6	6	6	30	26	27	32	35	30	31	-	1	0	1	0	2
32	4	3	3	3	1	6	6	9	8	8	6	3	1	1	2	5	5	5	4	5	6	6	6	7	2	2	1	1	3	3	2	1	-	2	0	0	0
33	3	1	3	2	1	1	0	3	6	2	2	1	1	0	0	8	5	6	7	6	4	3	2	2	2	1	1	1	3	1	2	0	2	-	0	1	0
34	0	0	0	0	0	0	1	1	0	0	2	0	1	1	0	2	1	1	1	1	1	2	1	2	0	2	1	1	1	1	1	1	0	0	-	18	9
35	0	0	0	0	1	1	1	0	0	1	1	0	2	1	0	2	1	0	0	2	2	1	1	1	0	0	0	0	0	0	0	0	0	1	18	-	16
36	0	0	0	0	0	1	2	0	0	0	1	0	0	2	0	1	1	1	1	0	1	0	0	1	0	1	1	1	1	1	1	2	0	0	9	16	-

Table 3: Feature Matches with high correspondence

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	-	26	18	13	17	15	18	19	23	20	17	13	9	16	7	19	21	22	21	18	30	32	30	27	6	9	5	6	11	9	9	8	10	7	9	12	10
1	26	-	23	12	13	11	18	16	18	14	17	13	7	11	9	22	22	20	22	22	26	26	24	28	6	8	7	7	10	11	9	9	10	9	8	11	7
2	18	23	-	33	15	12	18	15	20	17	14	14	9	10	10	17	19	21	23	20	24	26	28	27	10	15	10	9	12	12	13	11	11	7	7	15	10
3	13	12	33	-	20	7	10	12	14	15	8	13	6	8	7	15	15	15	17	16	19	20	24	23	5	8	5	5	9	7	6	6	7	7	7	9	7
4	17	13	15	20	-	10	14	12	17	11	13	12	7	8	6	14	13	13	15	17	16	17	17	14	6	8	7	6	10	6	8	8	6	7	3	10	8
5	15	11	12	7	10	-	18	13	14	11	7	13	12	11	12	10	9	12	12	13	17	13	19	17	10	10	7	11	11	8	9	9	14	6	10	10	9
6	18	18	18	10	14	18	-	13	19	13	16	17	10	10	6	20	18	18	21	19	18	22	24	26	11	11	9	14	13	5	10	11	9	9	8	9	11
7	19	16	15	12	12	13	13	-	12	10	11	16	14	12	10	15	13	10	16	15	19	21	21	21	11	11	9	7	11	10	11	12	10	5	7	12	14
8	23	18	20	14	17	14	19	12	-	15	13	23	13	9	13	20	19	17	19	17	19	26	22	24	10	10	11	10	10	9	11	12	10	6	6	12	12
9	20	14	17	15	11	11	13	10	15	-	19	19	12	9	12	10	11	14	13	12	19	19	24	19	9	10	10	10	12	9	10	12	12	11	8	14	11
10	17	17	14	8	13	7	16	11	13	19	-	17	10	12	14	15	15	15	14	13	25	29	23	21	13	10	8	9	11	9	10	12	11	9	11	11	9
11	13	13	14	13	12	13	17	16	23	19	17	-	8	11	8	19	13	17	18	12	25	22	26	23	7	11	6	9	9	7	8	7	12	11	9	14	10
12	9	7	9	6	7	12	10	14	13	12	10	8	-	16	20	13	14	12	15	14	21	22	21	19	12	6	12	9	9	8	10	9	11	5	6	7	5
13	16	11	10	8	8	11	10	12	9	9	12	11	16	-	16	15	18	18	16	16	22	17	22	20	9	9	10	7	9	9	10	11	13	6	6	9	5
14	7	9	10	7	6	12	6	10	13	12	14	8	20	16	-	10	6	7	8	7	16	17	16	15	6	11	10	7	12	7	8	7	11	4	9	9	6
15	19	22	17	15	14	10	20	15	20	10	15	19	13	15	10	-	11	15	17	12	35	33	32	30	5	3	2	6	4	4	7	5	11	5	11	15	14
16	21	22	19	15	13	9	18	13	19	11	15	13	14	18	6	11	-	15	11	14	31	30	26	27	6	5	4	6	6	8	8	5	10	6	12	15	13
17	22	20	21	15	13	12	18	10	17	14	15	17	12	18	7	15	15	-	9	17	33	39	26	32	5	5	4	4	4	4	8	5	9	8	14	20	19
18	21	22	23	17	15	12	21	16	19	13	14	18	15	16	8	17	11	9	-	10	30	31	31	24	6	4	2	5	7	6	7	7	12	5	13	20	13

Table 3 – Continued

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
19	18	22	20	16	17	13	19	15	17	12	13	12	14	16	7	12	14	17	10	-	29	28	25	27	4	4	4	5	8	6	4	6	9	6	9	15	14
20	30	26	24	19	16	17	18	19	19	19	25	25	21	22	16	35	31	33	30	29	-	35	27	43	14	12	14	13	18	15	16	16	18	11	19	24	26
21	32	26	26	20	17	13	22	21	26	19	29	22	22	17	17	33	30	39	31	28	35	-	33	34	13	13	13	16	17	13	12	13	18	12	16	26	22
22	30	24	28	24	17	19	24	21	22	24	23	26	21	22	16	32	26	26	31	25	27	33	-	31	14	16	11	17	19	13	12	12	19	13	17	24	24
23	27	28	27	23	14	17	26	21	24	19	21	23	19	20	15	30	27	32	24	27	43	34	31	-	15	18	14	14	16	15	14	14	19	12	19	25	23
24	6	6	10	5	6	10	11	11	10	9	13	7	12	9	6	5	6	5	6	4	14	13	14	15	-	11	15	10	8	14	7	9	7	3	6	7	7
25	9	8	15	8	8	10	11	11	10	10	10	11	6	9	11	3	5	5	4	4	12	13	16	18	11	-	15	9	13	12	12	11	11	6	4	8	4
26	5	7	10	5	7	7	9	9	11	10	8	6	12	10	10	2	4	4	2	4	14	13	11	14	15	15	-	9	9	10	13	10	10	6	5	8	5
27	6	7	9	5	6	11	14	7	10	10	9	9	9	7	7	6	6	4	5	5	13	16	17	14	10	9	9	-	7	14	10	7	12	5	6	7	6
28	11	10	12	9	10	11	13	11	10	12	11	9	9	9	12	4	6	4	7	8	18	17	19	16	8	13	9	7	-	9	9	12	10	5	7	9	8
29	9	11	12	7	6	8	5	10	9	9	9	7	8	9	7	4	8	4	6	6	15	13	13	15	14	12	10	14	9	-	14	12	9	4	4	6	7
30	9	9	13	6	8	9	10	11	11	10	10	8	10	10	8	7	8	8	7	4	16	12	12	14	7	12	13	10	9	14	-	10	9	6	7	7	6
31	8	9	11	6	8	9	11	12	12	12	12	7	9	11	7	5	5	5	7	6	16	13	12	14	9	11	10	7	12	12	10	-	9	6	5	8	5
32	10	10	11	7	6	14	9	10	10	12	11	12	11	13	11	11	10	9	12	9	18	18	19	19	7	11	10	12	10	9	9	9	-	5	6	10	7
33	7	9	7	7	7	6	9	5	6	11	9	11	5	6	4	5	6	8	5	6	11	12	13	12	3	6	6	5	5	4	6	6	5	-	4	2	3
34	9	8	7	7	3	10	8	7	6	8	11	9	6	6	9	11	12	14	13	9	19	16	17	19	6	4	5	6	7	4	7	5	6	4	-	16	21
35	12	11	15	9	10	10	9	12	12	14	11	14	7	9	9	15	15	20	20	15	24	26	24	25	7	8	8	7	9	6	7	8	10	2	16	-	22
36	10	7	10	7	8	9	11	14	12	11	9	10	5	5	6	14	13	19	13	14	26	22	24	23	7	4	5	6	8	7	6	5	7	3	21	22	-

Table 4: Feature Matches with average correspondence

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	-	16	22	21	18	16	17	13	23	21	19	17	27	19	19	25	26	27	33	21	30	35	35	38	12	15	15	19	14	7	14	11	17	8	17	20	21
1	16	-	21	27	32	22	19	17	17	17	18	13	26	23	19	19	16	25	18	17	35	32	35	31	12	11	12	14	10	7	11	11	15	6	15	17	22
2	22	21	-	31	30	18	12	13	17	15	20	14	23	22	16	21	25	22	19	17	30	34	30	32	11	10	14	13	12	10	10	13	17	9	15	13	15
3	21	27	31	-	31	17	19	18	17	18	19	15	24	20	14	29	26	30	27	19	32	32	25	28	11	9	10	16	9	10	9	11	19	9	14	16	18
4	18	32	30	31	-	17	15	21	19	21	16	14	21	18	11	23	28	26	27	18	32	31	33	37	12	12	11	15	9	12	12	10	18	8	24	17	17
5	16	22	18	17	17	-	7	9	14	12	16	11	14	17	12	12	12	13	12	13	19	22	21	18	12	8	10	12	10	12	15	16	11	7	15	16	21
6	17	19	12	19	15	7	-	10	14	14	18	12	18	18	17	16	16	14	11	15	24	21	20	17	8	9	10	8	11	12	12	14	14	7	19	19	14
7	13	17	13	18	21	9	10	-	9	19	16	13	11	17	15	17	15	16	9	16	18	15	17	18	10	8	5	11	10	5	7	7	6	8	18	22	17
8	23	17	17	17	19	14	14	9	-	9	22	18	19	20	12	15	17	20	11	18	26	18	26	24	16	13	14	15	12	9	13	12	18	10	21	26	22
9	21	17	15	18	21	12	14	19	9	-	7	9	11	18	10	17	15	15	12	13	18	22	17	17	13	10	7	10	11	8	9	8	11	6	15	13	12
10	19	18	20	19	16	16	18	16	22	7	-	13	16	17	11	14	14	16	10	11	20	15	17	21	10	7	7	7	7	7	8	9	13	9	14	24	17
11	17	13	14	15	14	11	12	13	18	9	13	-	13	10	13	16	13	12	10	14	13	18	13	13	14	11	9	11	14	11	11	13	10	8	12	14	13
12	27	26	23	24	21	14	18	11	19	11	16	13	-	21	18	20	19	25	15	18	23	23	21	27	13	14	10	17	13	13	13	15	13	6	16	13	16
13	19	23	22	20	18	17	18	17	20	18	17	10	21	-	19	20	15	20	17	20	28	37	27	29	14	10	9	13	11	9	10	10	17	11	22	14	18
14	19	19	16	14	11	12	17	15	12	10	11	13	18	19	-	11	14	14	12	12	28	25	23	26	11	5	3	8	6	7	8	9	8	9	14	14	15
15	25	19	21	29	23	12	16	17	15	17	14	16	20	20	11	-	4	8	7	13	27	27	26	26	7	9	16	13	12	11	11	14	12	7	22	22	18
16	26	16	25	26	28	12	16	15	17	15	14	13	19	15	14	4	-	5	7	7	25	25	30	28	8	8	14	13	12	11	9	13	9	7	22	19	19
17	27	25	22	30	26	13	14	16	20	15	16	12	25	20	14	8	5	-	4	10	25	22	32	24	10	9	11	14	12	10	9	10	13	3	24	14	14
18	33	18	19	27	27	12	11	9	11	12	10	10	15	17	12	7	7	4	-	11	21	28	24	25	11	13	15	14	12	12	7	10	12	5	19	14	22

Table 4 – Continued

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
19	21	17	17	19	18	13	15	16	18	13	11	14	18	20	12	13	7	10	11	-	26	29	31	20	11	8	9	11	9	7	9	10	16	7	22	16	16
20	30	35	30	32	32	19	24	18	26	18	20	13	23	28	28	27	25	25	21	26	-	26	33	30	13	12	11	16	12	14	13	12	19	7	31	23	23
21	35	32	34	32	31	22	21	15	18	22	15	18	23	37	25	27	25	22	28	29	26	-	19	27	12	12	13	12	12	14	14	11	22	8	33	24	31
22	35	35	30	25	33	21	20	17	26	17	17	13	21	27	23	26	30	32	24	31	33	19	-	23	13	10	13	11	13	18	16	18	24	9	32	26	26
23	38	31	32	28	37	18	17	18	24	17	21	13	27	29	26	26	28	24	25	20	30	27	23	-	9	8	12	14	9	10	13	14	22	11	30	20	24
24	12	12	11	11	12	12	8	10	16	13	10	14	13	14	11	7	8	10	11	11	13	12	13	9	-	10	6	7	5	6	10	8	14	7	10	10	12
25	15	11	10	9	12	8	9	8	13	10	7	11	14	10	5	9	8	9	13	8	12	12	10	8	10	-	5	9	6	9	5	9	8	3	9	8	12
26	15	12	14	10	11	10	10	5	14	7	7	9	10	9	3	16	14	11	15	9	11	13	13	12	6	5	-	4	4	2	7	7	14	4	10	10	9
27	19	14	13	16	15	12	8	11	15	10	7	11	17	13	8	13	13	14	14	11	16	12	11	14	7	9	4	-	5	5	7	7	15	4	12	8	12
28	14	10	12	9	9	10	11	10	12	11	7	14	13	11	6	12	12	12	12	9	12	12	13	9	5	6	4	5	-	4	6	5	10	4	10	8	10
29	7	7	10	10	12	12	12	5	9	8	7	11	13	9	7	11	11	10	12	7	14	14	18	10	6	9	2	5	4	-	7	7	11	6	16	8	8
30	14	11	10	9	12	15	12	7	13	9	8	11	13	10	8	11	9	9	7	9	13	14	16	13	10	5	7	7	6	7	-	5	14	3	10	10	10
31	11	11	13	11	10	16	14	7	12	8	9	13	15	10	9	14	13	10	10	10	12	11	18	14	8	9	7	7	5	7	5	-	15	4	10	8	8
32	17	15	17	19	18	11	14	6	18	11	13	10	13	17	8	12	9	13	12	16	19	22	24	22	14	8	14	15	10	11	14	15	-	11	18	22	25
33	8	6	9	9	8	7	7	8	10	6	9	8	6	11	9	7	7	3	5	7	7	8	9	11	7	3	4	4	4	6	3	4	11	-	7	10	7
34	17	15	15	14	24	15	19	18	21	15	14	12	16	22	14	22	22	24	19	22	31	33	32	30	10	9	10	12	10	16	10	10	18	7	-	21	17
35	20	17	13	16	17	16	19	22	26	13	24	14	13	14	14	22	19	14	14	16	23	24	26	20	10	8	10	8	8	8	10	8	22	10	21	-	23
36	21	22	15	18	17	21	14	17	22	12	17	13	16	18	15	18	19	14	22	16	23	31	26	24	12	12	9	12	10	8	10	8	25	7	17	23	-

Table 5: Inliers found using a threshold of 1 pixel

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	-	96	64	48	20	13	18	18	22	14	21	19	24	18	9	24	20	22	24	20	21	32	31	17	13	15	10	9	13	10	9	11	12	8	9	17	12
1	96	-	79	61	44	16	15	16	24	15	20	10	17	15	8	13	26	22	22	21	26	22	19	20	9	12	10	15	8	9	11	12	11	9	10	9	13
2	64	79	-	89	60	17	19	18	22	20	12	15	18	17	14	24	17	21	24	19	30	29	28	21	10	10	13	11	14	8	9	11	9	10	8	14	16
3	48	61	89	-	73	14	15	15	7	17	10	19	12	15	8	21	17	18	20	19	25	27	17	24	11	11	7	5	7	9	9	10	10	7	7	10	12
4	20	44	60	73	-	11	14	14	17	12	19	11	12	11	8	16	22	18	19	18	21	11	26	26	8	11	7	14	12	11	10	6	11	9	10	10	12
5	13	16	17	14	11	-	34	29	27	15	16	13	12	16	16	11	6	11	14	11	15	18	16	16	9	10	8	13	10	7	13	10	9	10	9	12	11
6	18	15	19	15	14	34	-	41	44	25	17	18	9	8	9	10	13	12	11	7	21	22	17	2	10	8	10	13	12	9	9	12	17	6	10	12	17
7	18	16	18	15	14	29	41	-	52	43	30	25	6	6	17	21	21	17	12	11	20	13	22	16	10	13	10	12	13	10	11	8	16	7	10	16	12
8	22	24	22	7	17	27	44	52	-	45	37	22	19	17	10	20	23	30	22	21	32	23	18	10	10	10	13	14	12	10	12	14	16	10	8	8	15
9	14	15	20	17	12	15	25	43	45	-	48	25	6	7	9	19	18	22	9	11	20	18	21	25	11	10	8	8	15	9	11	9	14	13	10	13	12
10	21	20	12	10	19	16	17	30	37	48	-	44	13	13	12	25	18	12	4	17	27	23	24	22	11	12	12	14	10	14	13	10	14	11	8	16	14
11	19	10	15	19	11	13	18	25	22	25	44	-	9	6	11	19	11	14	16	17	25	23	22	18	12	8	10	11	9	9	12	9	14	9	7	10	8
12	24	17	18	12	12	12	9	6	19	6	13	9	-	72	43	16	19	14	13	3	22	19	18	20	11	10	11	13	10	10	14	9	9	4	10	8	10
13	18	15	17	15	11	16	8	6	17	7	13	6	72	-	24	12	14	19	6	6	20	11	23	21	11	9	9	9	8	7	5	11	14	6	12	10	11
14	9	8	14	8	8	16	9	17	10	9	12	11	43	24	-	10	8	10	10	16	18	19	11	20	13	8	9	8	7	4	8	5	10	4	6	11	8
15	24	13	24	21	16	11	10	21	20	19	25	19	16	12	10	-	89	89	75	58	33	29	26	34	8	8	17	9	8	8	7	11	12	4	15	18	16
16	20	26	17	17	22	6	13	21	23	18	18	11	19	14	8	89	-	88	85	66	29	22	35	18	10	7	8	11	12	8	11	15	11	8	10	13	13
17	22	22	21	18	18	11	12	17	30	22	12	14	14	19	10	89	88	-	88	74	30	31	27	33	14	8	8	12	8	8	8	10	13	6	18	12	10
18	24	22	24	20	19	14	11	12	22	9	4	16	13	6	10	75	85	88	-	80	29	32	30	24	8	13	10	10	11	10	7	9	14	7	7	15	12

Table 5 – Continued

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
19	20	21	19	19	18	11	7	11	21	11	17	17	3	6	16	58	66	74	80	-	30	27	30	22	7	9	9	8	9	7	9	7	18	6	7	13	14
20	21	26	30	25	21	15	21	20	32	20	27	25	22	20	18	33	29	30	29	30	-	149	142	139	13	10	11	12	12	22	11	20	16	10	18	19	17
21	32	22	29	27	11	18	22	13	23	18	23	23	19	11	19	29	22	31	32	27	149	-	148	141	15	18	14	20	12	20	15	18	18	7	18	13	20
22	31	19	28	17	26	16	17	22	18	21	24	22	18	23	11	26	35	27	30	30	142	148	-	138	16	12	12	11	13	15	10	19	15	9	7	22	20
23	17	20	21	24	26	16	2	16	10	25	22	18	20	21	20	34	18	33	24	22	139	141	138	-	11	13	6	14	16	11	16	15	24	12	14	18	14
24	13	9	10	11	8	9	10	10	10	11	11	12	11	11	13	8	10	14	8	7	13	15	16	11	-	46	52	40	48	51	47	47	11	5	10	7	8
25	15	12	10	11	11	10	8	13	10	10	12	8	10	9	8	8	7	8	13	9	10	18	12	13	46	-	42	41	50	46	47	45	7	6	6	5	6
26	10	10	13	7	7	8	10	10	13	8	12	10	11	9	9	17	8	8	10	9	11	14	12	6	52	42	-	41	41	47	45	44	13	4	7	6	6
27	9	15	11	5	14	13	13	12	14	8	14	11	13	9	8	9	11	12	10	8	12	20	11	14	40	41	41	-	42	43	45	46	12	4	7	9	8
28	13	8	14	7	12	10	12	13	12	15	10	9	10	8	7	8	12	8	11	9	12	12	13	16	48	50	41	42	-	52	47	52	9	5	9	6	9
29	10	9	8	9	11	7	9	10	10	9	14	9	10	7	4	8	8	8	10	7	22	20	15	11	51	46	47	43	52	-	52	49	13	4	11	6	5
30	9	11	9	9	10	13	9	11	12	11	13	12	14	5	8	7	11	8	7	9	11	15	10	16	47	47	45	45	47	52	-	46	14	6	8	11	8
31	11	12	11	10	6	10	12	8	14	9	10	9	9	11	5	11	15	10	9	7	20	18	19	15	47	45	44	46	52	49	46	-	13	6	6	8	7
32	12	11	9	10	11	9	17	16	16	14	14	14	9	14	10	12	11	13	14	18	16	18	15	24	11	7	13	12	9	13	14	13	-	9	10	12	14
33	8	9	10	7	9	10	6	7	10	13	11	9	4	6	4	4	8	6	7	6	10	7	9	12	5	6	4	4	5	4	6	6	9	-	4	3	5
34	9	10	8	7	10	9	10	10	8	10	8	7	10	12	6	15	10	18	7	7	18	18	7	14	10	6	7	7	9	11	8	6	10	4	-	46	44
35	17	9	14	10	10	12	12	16	8	13	16	10	8	10	11	18	13	12	15	13	19	13	22	18	7	5	6	9	6	6	11	8	12	3	46	-	55
36	12	13	16	12	12	11	17	12	15	12	14	8	10	11	8	16	13	10	12	14	17	20	20	14	8	6	6	8	9	5	8	7	14	5	44	55	-

PERMISSION TO COPY

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Texas Tech University or Texas Tech University Health Sciences Center, I agree that the Library and my major department shall make it freely available for research purposes. Permission to copy this thesis for scholarly purposes may be granted by the Director of the Library or my major professor. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my further written permission and that any user may be liable for copyright infringement.

Agree (Permission is granted.)

Sivabalan Muthukumar May 4, 2005

Disagree (Permission is not granted.)

_____- _____