Camera Rotation Recovery from Vanishing Point Matching

by

John Chung Hon Leung
B.A.Sc., University of Waterloo, 1993

A Thesis Submitted in Partial Fulfilment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Mechanical Engineering

We accept this thesis as conforming
to the required standard

[redacted]

Dr. G.F. McLean, Supervisor (Department of Mechanical Engineering)

[redacted]

Dr. N. Lind, Supervisor (Department of Mechanical Engineering)

[redacted]

Dr. C. Konzelman, Departmental Member (Department of Mechanical Engineering)

[redacted]

Dr. P. Agathoklis, Outside Member (Department of Electrical Engineering)

[redacted]

Dr. K.O. Niemann, External Examiner (Department of Geography)

University of Victoria

Supervisor: Dr. G.F. McLean, Dr. N. Lind

Abstract

Recovering camera rotations is an important operation of any stereo vision system and photogrammetric work. This thesis examines the method of recovering camera rotations using matched vanishing points between two images. Due to imperfection of automated vanishing point detection algorithms, natural line intersections can be mistaken as vanishing points. These false vanishing points can cause severe difficulties in automated vanishing point matching algorithms. This thesis examines the use of an invariant property exhibited by a group of true vanishing points to perform the matching. In this thesis, relaxation labelling is used to match vanishing points. The method performed successfully recovering camera rotations from matched vanishing points. The method was also useful in discriminating true and false vanishing points resulting from the imperfection of automated vanishing point detection techniques.

Examiners:

[redacted]

Dr. G.F. McLean, Supervisor (Department of Mechanical Engineering)

[redacted]

Dr. N. Lind, Supervisor (Department of Mechanical Engineering)

[redacted]

Dr. C. Konzelman, Departmental Member (Department of Mechanical Engineering)

[redacted]

Dr. P. Agathoklis, Outside Member (Department of Electrical Engineering)

[redacted]

Dr. K.O. Niemann, External Examiner (Department of Geography)

Table of Contents

List of Tables

List of Figures

Acknowledgement

I would like to thank Dr. McLean, my supervisor, for his assistance in the preparation of this thesis. I would also like to thank Dr. Lind and Dr. McLean for their financial support. Thanks also to Geoff Vanderkooy and Jason Szabo for their discussion, encouragement, and editing assistance in the preparation of this thesis.

For My Parents

1. Introduction

In the field of photogrammetry, one wishes to obtain three-dimensional information, such as dimensions or a solid model of an object, from two-dimensional images. Unfortunately, the information from one of the dimensions is lost because of the perspective projection, to be discussed in Chapter 2. To recover the third dimension (sometimes called the depth information) requires at least two images of the same scene taken from different view points. This is the same reason why the human vision system requires the use of two eyes in order to estimate the distance to an object.

This process of recovering three dimensional information from two dimensional images can be broken down into five stages: camera calibration, feature extraction, feature correspondence, reverse projection, and model reconstruction. The purpose of camera calibration is to obtain a geometric model of the camera which describes the relationship between the camera's and the object's coordinate systems. Details of this subject will be discussed in Chapter 2. Feature extraction and feature correspondence refer to extracting some specific features in the image, such as lines or corners, and then locating the corresponding features in the second image. With knowledge of feature correspondence, the camera's geometric model, and the camera's location, the actual location of the feature in three dimensions can be computed. This process is known as reverse projection. The last stage involves connecting these features in three dimensions into something more meaningful such as a solid model or line drawing.

Most work on feature correspondence requires knowledge of the camera's geometric model and its location. The epipolar constraint [1], for example, make use of this knowledge to constrain the search space from the entire image into a small strip in the image. In order to perform the task of reverse projection, knowledge of the camera's geometric model and its location are also required. Given the importance of camera calibration, this work concentrates on one aspect of this process, recovering the relative orientation of the camera between two images.

Various approaches to recover camera orientation will be examined in Chapter 2. Most require either specialized equipment and calibration patterns or manual assistance to perform this calibration. They also require the user have a good knowledge of stereo vision and photogrammetry. This thesis proposes a technique that can be fully automated and unsupervised. The overall goal is to allow a user to freely move the camera around a man-made monument, take pictures, and have the computer automatically extract the three dimensional model of the monument without human supervision. This goal calls for an automated system to recover camera orientation without modifying the scene (that is, no calibration patterns or modification of the environment is allowed).

One approach that can satisfy these criteria involves vanishing point matching. The vanishing point information between two images can be used to recover camera orientation. However, due to the complications in automated vanishing point detection, erroneous vanishing points will sometimes exist. These erroneous vanishing points cause

significant problems in the vanishing point matching process. Three different approaches to handle this erroneous identification of vanishing points in the matching process have been examined in this thesis.

The background of camera orientation recovery will be discussed first in Chapter 2. Due to the importance of the vanishing point matching approach, background on vanishing points will be discussed in Chapter 3. Three approaches to match vanishing points between two images will be presented in Chapter 4. Experimentation on the three approaches will then be discussed in Chapter 5, followed by the conclusions and recommendations in Chapter 6.

2. Recovering Relative Camera Orientation

Solving relative camera orientation is part of the process commonly known as camera calibration. Normally, the calibration involves solving the internal geometrical properties of the camera (intrinsic parameters) and the camera's location relative to a reference or global coordinate system (extrinsic parameters). The intrinsic parameters are the focal length, principal point of the image, and the scale factors between the image plane coordinate system and the global coordinate system. The extrinsic parameters are the translation and rotation of the camera with respect to the global coordinate system.

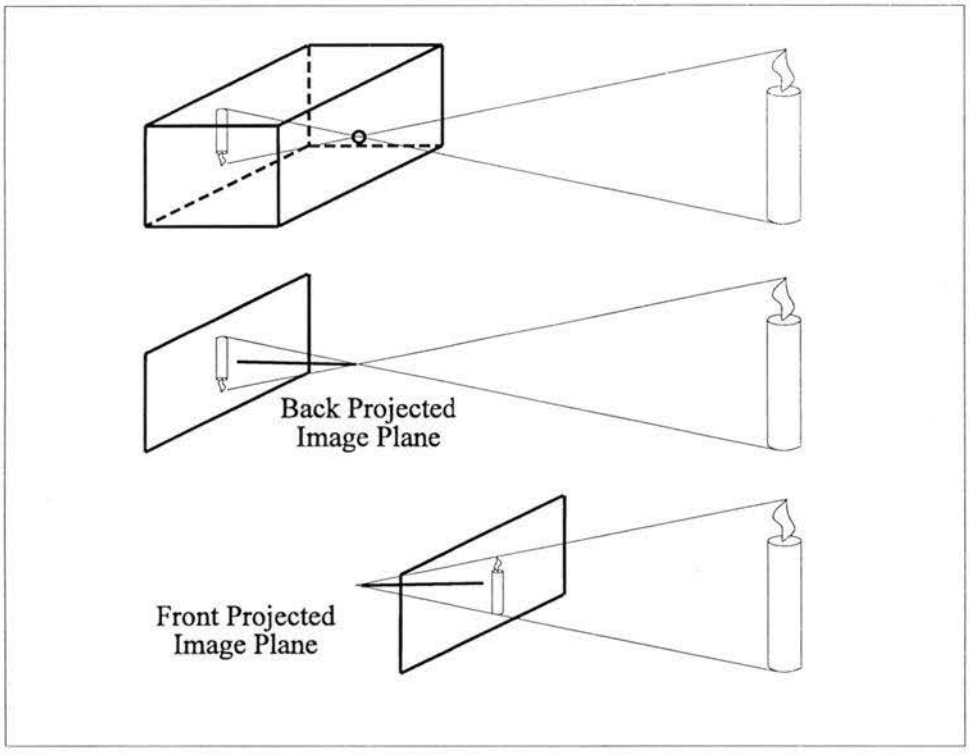


Figure 1 Pinhole Camera Model

Figure 1 illustrates the pinhole camera model where a single light ray, emitted or reflected from an object, goes through a pinhole and then hits the image plane. The resulting image is always inverted. The closest distance between this pinhole and the image plane is commonly called the focal length. The optical axis is the axis that goes through the pinhole (or the centre of projection) and intersects perpendicular to the image plane. This intersection is called the principal point or the optical centre of the camera. Sometimes, for convenience and simplicity, the image plane is placed between the real object and the centre of projection. The advantage of this representation is that the image is no longer inverted. In this thesis, the front projection system will be used to represent the camera.

Using the pinhole camera model shown in Figure 1 and the coordinate system shown in Figure 2, the relationship between the image point and the real-world three-dimensional point can be expressed as:

$$\frac{f}{z_c} = \frac{x_i}{x_c} = \frac{y_i}{y_c} \tag{1}$$

This perspective projection transformation can be represented using homogeneous coordinates:

$$\begin{bmatrix} w x_i \\ w y_i \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \tag{2}$$

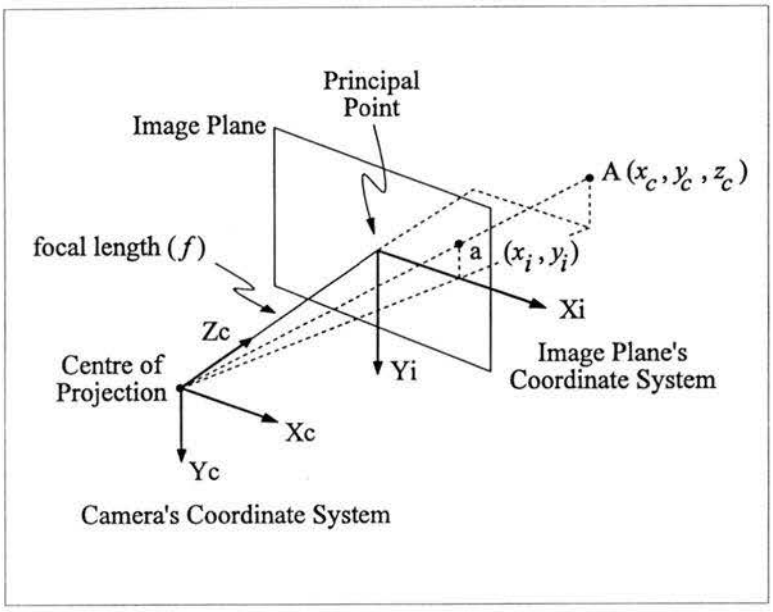


Figure 2 Perspective Projection

This assumes that the image plane's coordinate system origin is located at the principal point and that the scale between the two coordinate systems is the same. To take into account that the image plane coordinate system's origin could be located somewhere other than the principle point, such as the image's corner, and that there could be a difference in scale between the two coordinate systems, the perspective projection transformation can be expanded as [1]:

$$\begin{bmatrix} w\,x_i \\ w\,y_i \\ w \end{bmatrix} = \begin{bmatrix} fs_x & 0 & c_x & 0 \\ 0 & fs_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \tag{3}$$

where (c_x, c_y) is the location of the principal point expressed in the image plane coordinate system while s_x and s_y are the scale factors between the two coordinate systems in x and y directions, respectively.

If the real world point is expressed in terms of some global coordinate system, instead of the camera's coordinate system, then knowledge of the camera's position and orientation with respect to this global coordinate system is required. Therefore the real world point will have to be converted into the camera's coordinate system using homogeneous translation and rotation matrices [2]. Then the complete perspective transformation of a real world point to an image point is:

$$\begin{bmatrix} w\,x_i \\ w\,y_i \\ w \end{bmatrix} = PRT \begin{bmatrix} x_g \\ y_g \\ z_g \\ 1 \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \end{bmatrix} \begin{bmatrix} x_g \\ y_g \\ z_g \\ 1 \end{bmatrix} \tag{4}$$

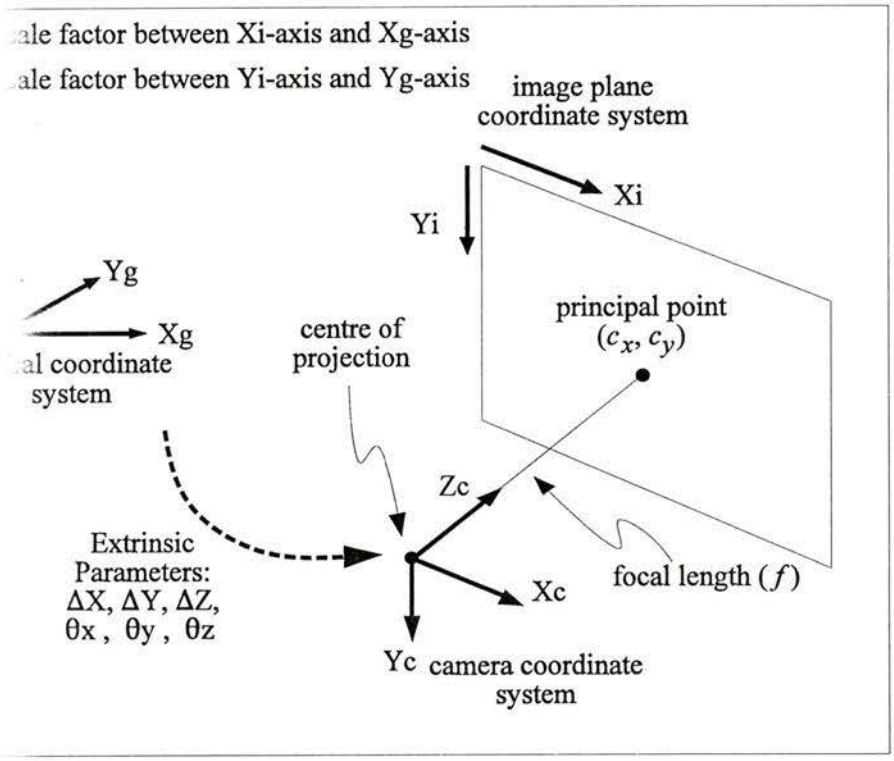
where P is the 3-by-4 perspective transformation matrix shown in Equation (3), R is a composite of three 4-by-4 homogeneous rotation matrices in the order of rotation about x-axis (θ_x), then rotation about y-axis (θ_y), then rotation about z-axis (θ_z):

$$R = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 & 0 \\ \sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & \sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

and T is a 4-by-4 homogeneous translation matrix:

$$T = \begin{bmatrix} 1 & 0 & 0 & \Delta X \\ 0 & 1 & 0 & \Delta Y \\ 0 & 0 & 1 & \Delta Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

nation involves 5 intrinsic camera parameters (f, c_x, c_y, s_x, s_y) and 6 extrinsic eters (θ_x, θ_y, θ_z, Δx, Δy, Δz) in order to fully describe the relationship orld point and the image point. Figure 3 summarizes these camera a pinhole camera model. In this thesis, the camera will be represented by and an image plane as shown in the figure.



3 Camera Parameters

reo vision systems, the extrinsic parameters are sometimes expressed in the ive translation and rotation between two cameras (see Figure 4) [3, 4, 5]. insic parameters represent the internal geometry of the camera, they will nt as the camera moves from one position to the next. However, because

the extrinsic parameters are related to the camera positions, they must be re-computed if the camera is moved. A method that can automate the process of determining the extrinsic parameters every time the camera is moved is desired to speed up this time-consuming re-calibration.

There are various methods of recovering relative orientation between two cameras. This chapter examines these methods and their suitability to meet the overall goal as defined in the Introduction.

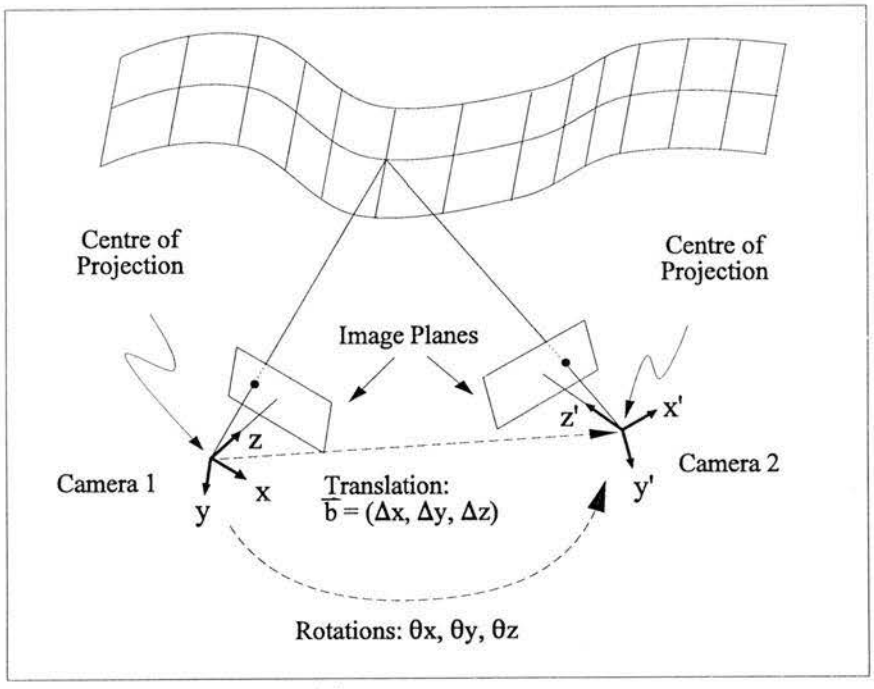


Figure 4 Stereo Visions

2.1 Stereo Rigs and Calibration Pattern

One approach to this problem is to rigidly connect two cameras to form a "stereo rig" and then calibrate the two cameras separately using calibration patterns [1, 6, 7, 8]. The calibration pattern could be any regular grid pattern such as a checkerboard. Placing two patterns perpendicular to each other allows convenient location and measurement of the three dimensional points at the corners of each grid in the pattern. The corresponding two dimensional image points can be located manually. Rewriting Equation (4) produces two equations:

$$\begin{aligned} \left(\left[q_{31}\ q_{32}\ q_{33} \right] \left[x_g\ y_g\ z_g \right]^T + q_{34} \right) x_i &= \left[q_{11}\ q_{12}\ q_{13} \right] \left[x_g\ y_g\ z_g \right]^T + q_{14} \\ \left(\left[q_{31}\ q_{32}\ q_{33} \right] \left[x_g\ y_g\ z_g \right]^T + q_{34} \right) y_i &= \left[q_{21}\ q_{22}\ q_{23} \right] \left[x_g\ y_g\ z_g \right]^T + q_{24} \end{aligned} \tag{7}$$

Since each pair of point correspondences provides 2 equations with 11 unknown parameters, a minimum of 6 real world to image point pairs allows the camera intrinsic and extrinsic parameters to be computed.

Although this approach is simple and easy to implement, it requires the use of two cameras in a stereo rig and a calibration pattern to achieve the task. Furthermore, obtaining more measurements of the object where the surface is not visible from the two original camera positions requires that the stereo rig be moved around the object. This effectively means that the cameras need to be re-calibrated again.

2.2 Point Correspondence Between Images

Some other methods used to recover relative camera orientation and position make use of point correspondence between two images [3, 4, 5]. Unlike the stereo rig discussed in Section 2.1, these methods use image-to-image point correspondence instead of real world to image point correspondence. The location of the real world points is unknown, but the location of two corresponding image points (x_{i1}, y_{i1}) and (x_{i2}, y_{i2}) is. The global coordinate system is defined by the first camera's coordinate system, so the extrinsic camera parameters represent the relative translation and rotation between the two cameras. Simplifying the perspective transformation in Equation (3) for the two cameras gives:

$$\begin{aligned} x_{ik} &= \left(\frac{f}{z_{ck}} x_{ck} \right) s_x + c_x \\ y_{ik} &= \left(\frac{f}{z_{ck}} y_{ck} \right) s_y + c_y \end{aligned} \qquad k = 1, 2 \ \textit{for camera 1 and 2} \tag{8}$$

and the relationship of a particular real world point expressed in the two camera coordinate systems (x_{c1}, y_{c1}, z_{c1}) and (x_{c2}, y_{c2}, z_{c2}) is:

$$\begin{bmatrix} x_{c2} \\ y_{c2} \\ z_{c2} \\ 1 \end{bmatrix} = R\ T \begin{bmatrix} x_{c1} \\ y_{c1} \\ z_{c1} \\ 1 \end{bmatrix} \tag{9}$$

where R and T are the rotation and translation matrices shown in Equations (5) and (6),

respectively. Note that the absolute distance, $d = \sqrt{\Delta X^2 + \Delta Y^2 + \Delta Z^2}$, between the two cameras is not constrained in any other equations, therefore it is not possible to solve for it and only a scaled model (proportional to the real system) can be obtained. This also means that now there are only five extrinsic parameters (three for rotations and two for translation). Each point correspondence produces 4 equations with 3 unknowns (x_c, y_c, z_c). In addition there are the 5 unknown extrinsic camera parameters to solve for. Therefore, a minimum of five point correspondences are required to solve for the extrinsic camera parameters. In practice, more than five point correspondences are used and the constraint equations are re-formulated so that a least square error analysis can be performed and the solution can be solved through iterative methods [3, 4, 5]. This approach is called bundle adjustment.

The problem with this approach is the requirement for image point correspondence to be found, implying an algorithm which can automatically detect and match image points from one image to the next. Paradoxically, most automated point correspondence algorithms rely on the knowledge of the two camera positions and orientations to reduce the search space of the point from the entire image to a small strip in the image. This leads back to the original problem of recovering the camera extrinsic parameters in order to perform the point correspondence.

2.3 Mobile Robots and Feature Tracking

Visually guided mobile robots frequently encounter the same problem of recovering camera rotation. In this problem the camera movement is small between two image frames, and therefore any image feature will not travel far away from its position in the previous image frame. Tracking the image feature movement becomes more feasible by examining a localized region of the image based on its previous position. With enough image points being matched between successive frames, the relative camera rotation problem can be solved. Also, with the knowledge of the robot velocity feedback, an initial estimate of the camera position for the iteration scheme can be provided, resulting in a quicker convergence.

This approach seems inappropriate for the overall goal of allowing a human user to freely move the camera. In this case, the camera rotation between two successive images will not be small. Furthermore, determination of camera motion will not be possible. However, the technique used by Shigang *et al.* [9] deserves special attention. Their technique is unique in that they chose the vanishing points of lines on horizontal planes as features. A vanishing point is the intersection of a set of parallel lines projected onto the image plane (see Figure 5). Railroad tracks, for example, when extended towards the horizon will be perceived by the human eyes as converging into a point. Vanishing lines, on the other hand, are the projected intersections of parallel planes (see Figure 6). The horizon, for example, is a vanishing line of different horizontal planes. If

different sets of parallel lines are also parallel to the horizontal plane, then their projected vanishing points will lie on the vanishing line (the horizon). By tracking the movement of these vanishing points on the horizon, camera rotation can be recovered.

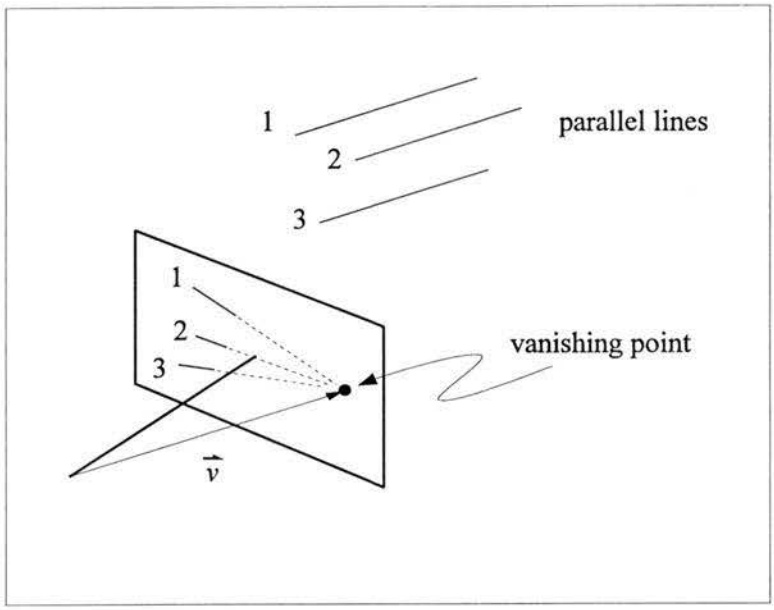


Figure 5 Vanishing Point

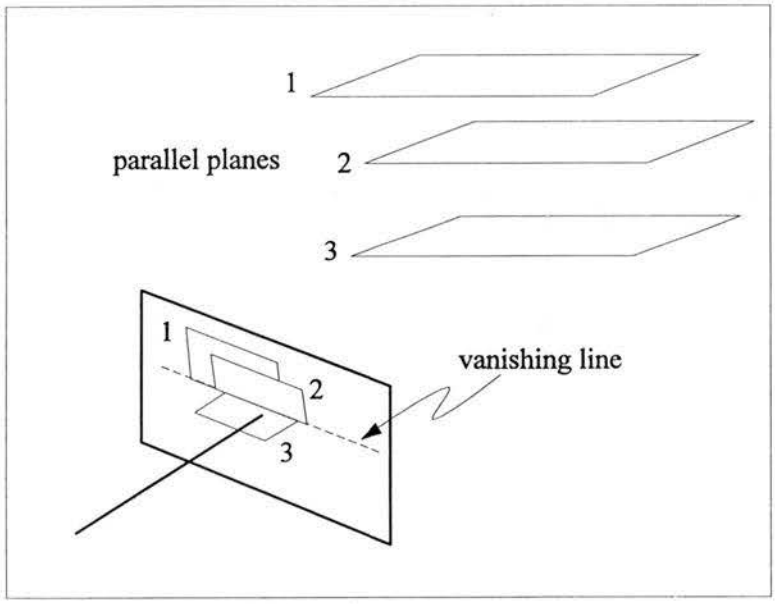


Figure 6 Vanishing Line

Although this approach requires numerous constraints (such as knowledge of robot velocity and the restriction of robot movement on a flat horizontal floor), it introduced the important concept of recovering camera rotation from vanishing point correspondence.

2.4 Camera Calibration Using Vanishing Points

The concept of using vanishing points for camera calibration has been examined by various researchers [7, 8, 10, 11]. However, many of these methods require the use of a calibration pattern. Chen and Jiang [7], for example, used grid paper to provide the vanishing points. Once the horizontal and vertical vanishing points are extracted from the image, the orientation of the grid lines with respect to the camera coordinate system can be determined[1]. Since the orientation of these grid lines with respect to a global coordinate system can be measured manually, the comparison between these computed and measured orientations allowed the camera rotations with respect to the global coordinate system to be recovered.

Wang and Tsai's [8] technique is another example that uses vanishing point information from a calibration target to extract the camera parameters. The target they

---

[1]Details on the relationship between vanishing points and the orientations of the parallel lines in three-dimensional space will be presented in Chapter 3.

used is a flat hexagon shape where its opposite sides are parallel to each other to provide three vanishing points. Because the three pairs of parallel lines lie on the same plane, the extracted vanishing points will lie on a vanishing line. Wang and Tsai discovered a relationship between the camera orientations and the geometry (orientation and location) of these vanishing points and the vanishing line. Camera orientations can be recovered using this relationship.

Caprile and Torre [10] used a set of parallel lines on a planar surface, together with a baseline that is perpendicular to the parallel line set, as a calibration target. For three mutually orthogonal straight lines in space, with their corresponding vanishing point vectors ($\vec{v}$ in Figure 5), if the location of one vanishing point is known and the direction of an image line that passes through a second vanishing point is also known, then the locations of the remaining two vanishing points can be computed. Applying this property on their calibration target, they extracted three mutually orthogonal vanishing point vectors from the image. Then solving the relative camera rotation between two images becomes solving the rotation matrix that is responsible for rotating the three mutually orthogonal vanishing point vectors from one image to the next:

$$R = V_2 V_1^{-1} = V_2 V_1^T \tag{10}$$

where the columns of V_i are the three mutually orthogonal vanishing point vectors in each image.

Although the above approaches provided a means of recovering camera orientations by using vanishing points, the techniques still require the use of calibration targets to achieve the goal. These techniques are acceptable for indoor laboratory environments where the camera is expected to be calibrated once and is not intended to be moved again. However, for outdoor applications, photographers do not always have the luxury of placing calibration targets anywhere they wish. For example, if the top portion of a building is being measured, it would be difficult to place a reasonable sized target in the scene. Note that if the target is too small, the resulting image of the target will also be too small to provide any accurate measurement. Due to these problems, it would be more desirable to use vanishing point information obtained directly from the environment instead of using calibration targets.

Quan and Mohr's [11] work uses vanishing points extracted directly from the environment instead of some calibration object. They assumed that there will be at least two vanishing points in the images and that the rotational motion of the camera is not too large. Recovering camera rotations then consists of finding the corresponding vanishing points between two images and tracking the movement of the vanishing point vectors between two images. Because of the assumption of small camera rotation, the vanishing point correspondence problem between two images is simply solved by finding the nearest vanishing point from its previous location. This approach is very appealing because it does not require special equipment like the stereo rig, calibration pattern, or mobile robot. However, the next chapter discusses the difficulty of correctly identifying

a vanishing point in an image. The presence of erroneous identifications would seriously affect this simple matching process, resulting in incorrect camera rotation estimation.

2.5 Summary

Four general approaches to recover relative camera orientations were reviewed: calibration objects, point correspondence, mobile robots, and vanishing point matching. The use of calibration objects require the cameras to be fixed at one location. Camera movement requires re-calibration. Furthermore, for non-laboratory environments, it is not always possible to place the calibration objects in the scene. Therefore, this approach is undesirable if free camera movement in an outdoor environment is allowed. The point correspondence technique generally requires point matching to be performed manually. This violates the goal of a completely automated process. The use of a mobile robot violates the criterion of allowing human users to freely move the camera and take pictures of a man-made monument as they wish. Among the four general methods being discussed, only the vanishing point matching approach can effectively allow extraction of information from the environment to recover camera rotations in an automated fashion. Because of the appeal of this approach, it was decided to further investigate vanishing point matching techniques in this thesis.

In the next chapter, the basic properties of vanishing points and some of the common methods in detecting vanishing points in an image will be discussed. The problem with erroneous vanishing point identification associated with automated vanishing point detection will also be examined.

3. Vanishing Points

Before attempting to match vanishing points between two images, it is crucial to understand the properties of vanishing points and the methods of vanishing point detection. This chapter will first examine the basic properties of vanishing points. Then problems and existing solutions to vanishing point detection will be examined. The method of recovering camera rotation using vanishing points will also be discussed.

3.1 Vanishing Point Representation

Under perspective projection, lines which are parallel in space will converge to a point on the image plane. Such points are called vanishing points. Because it is a point on the image plane, it can naturally be expressed in terms of the image plane coordinate system (Xi, Yi). However, if the camera is aiming perpendicular to the set of parallel lines in space, this set of lines will also be parallel on the image plane. In this circumstance, the vanishing point location cannot be defined because by definition parallel lines do not intersect. Therefore, vanishing points are also very often expressed in terms of the Gaussian sphere [12], which is a unit sphere located at the centre of projection of the camera (Figure 7). Using this representation, a vanishing point becomes a vector ($\vec{v}$) pointing towards the location of the vanishing point on the image plane and

parallel to the lines in space. In the above case of parallel lines on image plane, the vanishing point becomes a unit vector that is parallel to the set of parallel lines in space. Using this representation scheme, an unbounded image plane is now transformed to the bounded surface of a sphere. In the following discussion, the term vanishing point vector refers to this unit vector direction or equivalently a point on the Gaussian sphere.

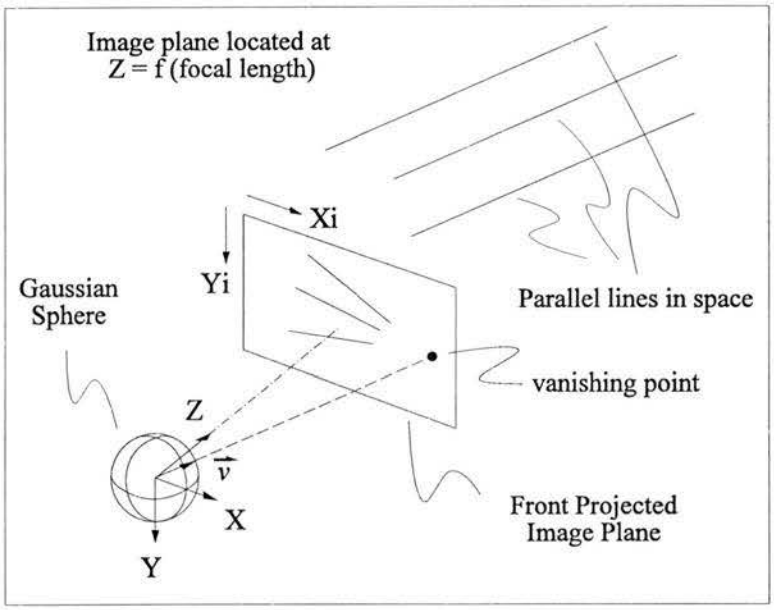


Figure 7 Gaussian Sphere

3.2 Vanishing Point Properties

In this section, the properties associated with vanishing points that are crucial to vanishing point matching will be examined, including the geometry of the vanishing point vector, translation invariance, invariant structure, and conjugate vanishing points.

3.2.1 Vanishing Point Vector Orientation

Although a vanishing point is defined as the intersection of parallel lines under perspective projection, it can also be viewed as the projection of a point on the line at infinity [10, 12]. Figure 8 illustrates that as a point on the line moves further out towards infinity, the orientation of the projection ray becomes more parallel to the lines in space. Ultimately, the orientation of this ray will reach a limit which is exactly parallel to the parallel line set. This limiting ray also defines the location of the vanishing point. This shows that the orientation of vanishing point vectors are parallel to their corresponding sets of parallel lines. A more rigorous proof of this property is shown in Appendix A.

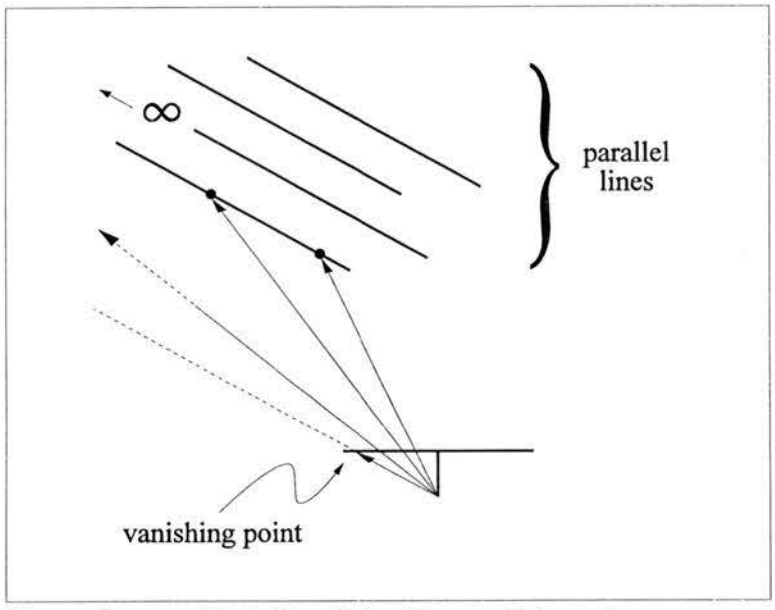


Figure 8 Vanishing Point Vector Orientation

3.2.2 Translation Invariance

Since the vanishing point vector is parallel to the corresponding line set in space, the location of this vanishing point is invariant under camera translation. From the proof in Appendix A, the location of any vanishing point expressed in terms of the camera coordinate system can be written as:

$$\vec{p}_{vp} = \left[\frac{f}{d_z} d_x \quad \frac{f}{d_z} d_y \quad f \right]^T \tag{11}$$

where f is the camera's focal length and vector $\vec{d} = [\, d_x \;\; d_y \;\; d_z \,]$ is the direction of the line in space. This is not a function of the location of the line, but of the orientation of the line and the focal length. Therefore, the location of vanishing points will remain the same regardless of camera translation. This invariant property is the main reason why although it is possible to recover camera rotation, it is impossible to recover camera translation from matched vanishing points between two images.

3.2.3 Invariant Constellation Structure

A group of vanishing points on the Gaussian sphere forms a relational structure which is invariant to camera rotation. This structure will be referred to as a constellation. This property can be explained through Figure 9. The vanishing point vector is parallel to the parallel line set in space. Since a normal solid object does not change shape when the camera is moved, the group of vanishing points associated with this object will not change shape either, thus producing this invariant property. A proof of this property is shown in Appendix B. Note that erroneous vanishing point identifications, such as at the

corner of a building being wrongly identified as a vanishing point, will not exhibit this invariant property. Therefore, constellation invariance becomes the key component for vanishing point matching when erroneous vanishing point identification exists in the data set. Also, estimating camera rotations reduces to recovering the rotation of the constellation.

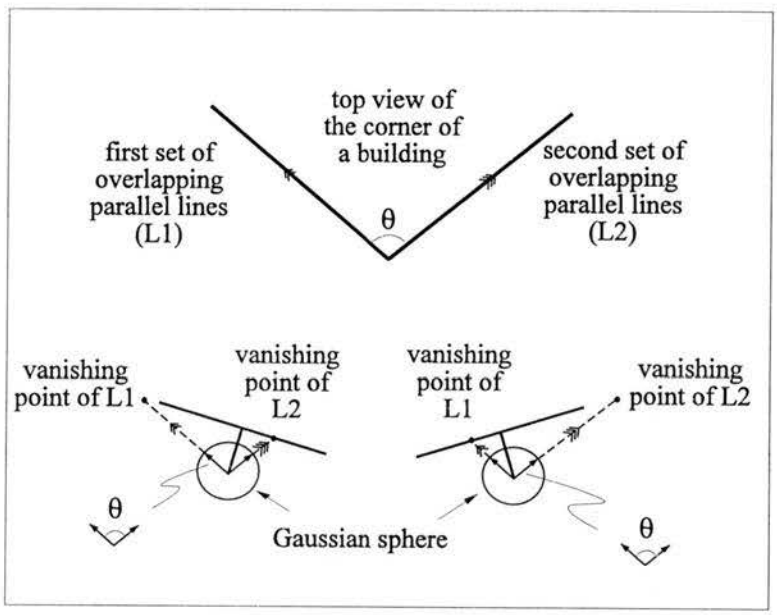


Figure 9 Invariant Vanishing Point Structure

3.2.4 Conjugate Vanishing Points

Recall that a vanishing point is also the projection of the intersection of parallel lines at infinity. Then there are always two vanishing points associated with the set of parallel lines because there are always two endpoints for any line. These two vanishing points are 180° from each other on the Gaussian sphere and they will be referred to as

conjugate vanishing points. The effect of conjugate vanishing points becomes apparent when the camera is nearly perpendicular to the parallel line set. As shown in Figure 10, when the camera is at position 1, the vanishing point is located at the right side of the image. However, after the camera is moved to position 2, the vanishing point associated with the same set of lines suddenly moves to the left side of the image. The vanishing point vectors themselves are pointing directly opposite to each other. Therefore, the vanishing point vector from camera position 2 is the conjugate pole of the one in camera position 1. When tracking the movement of vanishing points to recover camera orientation, it is important to realize which pole is being matched between two camera positions. Mis-identifying the pole could lead to incorrect camera rotation recovery.

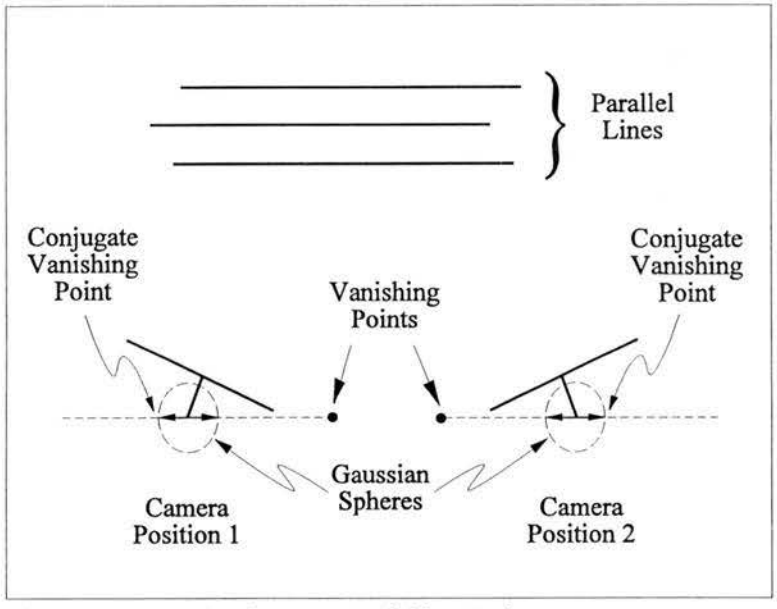


Figure 10 Conjugate Vanishing Point

3.3 Vanishing Point Detection

Because vanishing points are composed of intersections of lines on the image plane, detecting vanishing points naturally involves some form of clustering of line intersections. The majority of the detection methods make use of the accumulator space techniques to perform clustering. Barnard [12], for example, suggested a method to map the image lines onto the Gaussian sphere forming Great Circles[2]. Then line intersections on the image plane now becomes Great Circle intersections. To obtain these clusters of intersections, the Gaussian sphere is parameterized into azimuth and elevation[3] (angular representation of a point on a sphere). Then Great Circles are traced out on this parameterized accumulator space and local maximas, representing the potential vanishing point locations, are extracted. The purpose of this complicated Gaussian sphere mapping is to provide a bounding limit for the accumulator space. Quan and Mohr [13] used a similar Hough transform[4] [14] technique except that they used a hierarchical accumulator space. Since the straightforward parameterization of the Gaussian sphere will lead to an irregular division of the surface of the sphere, the problem of location accuracy arises when the bin size at the equator of the sphere is too coarse. Quan and Mohr's work

---

[2]A plane can be defined by a line and a point. Using an image line and the centre of projection, a plane is defined. When this plane intersects the Gaussian sphere, a circle can be traced out on the sphere and this circle is called the Great circle.

[3]Azimuth is the angle measured from the camera's z-axis in the x-z plane while elevation is the angle measured from the camera's x-z plane towards the y-axis.

[4]In general, Hough transform means a mapping of a feature space into a parameter space such as the mapping of image plane (x-y space) into the parameterized Gaussian sphere (azimuth and elevation).

divided the coarse bins into finer bins recursively and therefore provided better accuracy in vanishing point location.

To improve the location accuracy of vanishing points, Magee and Aggarwal [15] used a series of cross-product operations to determine the vanishing point vectors directly and then parameterized these vectors into the azimuth and elevation representation of the Gaussian sphere mapping. Then they perform cluster analysis on all line intersections using arc distance between two vanishing point vectors as the measure metric. Because this method used a continuous parameterization of the Gaussian sphere instead of the non-uniform discrete parameterization, the estimated vanishing point location is greatly refined.

The detection algorithms described above possess one potential problem. Line intersection clusters are not necessary all vanishing points. Very often, there are many naturally formed line intersection clusters in an image, such as the corner of a building or an array of windows. These natural intersection clusters could potentially confuse the detection algorithm and be wrongly identified as vanishing points. There are a few other methods that attempt to estimate the true vanishing points by using a probabilistic approach. Brillault-O'Mahony [16], for example, designed an accumulator space that uniformly distributed the probability of erroneous vanishing point detection. The basic idea is that detected line end-points from an image may be in error due to the results of digitization and image processing. As a result, it creates an uncertainty for the line

segment passing through the vanishing point. Brillault-O'Mahony developed a statistical model to transform the image plane into an accumulator space such that the uncertainty is constant throughout the accumulator space. However, the drawback to this approach is that the method is still based on the use of accumulator space to obtain vanishing points which therefore limits its accuracy.

The latest technique that uses statistical reasoning to detect vanishing points is the one developed by McLean and Kotturi [17]. Their method integrates the information from image processing into vanishing point detection. In the step of line detection, a measure of line quality was extracted. Using this information, an error model is derived to create a criterion for line clustering directly on the image plane. Once the lines associated with the same vanishing point are clustered into one group, the actual location is determined through optimization.

Unfortunately, with all of these probabilistic techniques to better estimate vanishing points, the underlying problem still exists. Corners of buildings are still naturally formed line intersections. When those lines are projected onto an image plane, they will still form a perfectly valid intersection. Therefore, there is always the possibility of false vanishing point identification. Because of the existence of these erroneous identifications, the process of matching vanishing points between two images becomes more complicated.

3.4 Computing Camera Rotations Using Vanishing Points

Computing the relative camera rotation using vanishing points requires at least two matched pairs from two images. With one pair of matched vanishing points, the possible location of the axis of rotation is constrained to be on a plane that bisects the two matched vanishing points (Figure 11). Then with two pairs of matched vanishing points, the axis of rotation can be located by intersecting the two bisecting planes (Figure 12).

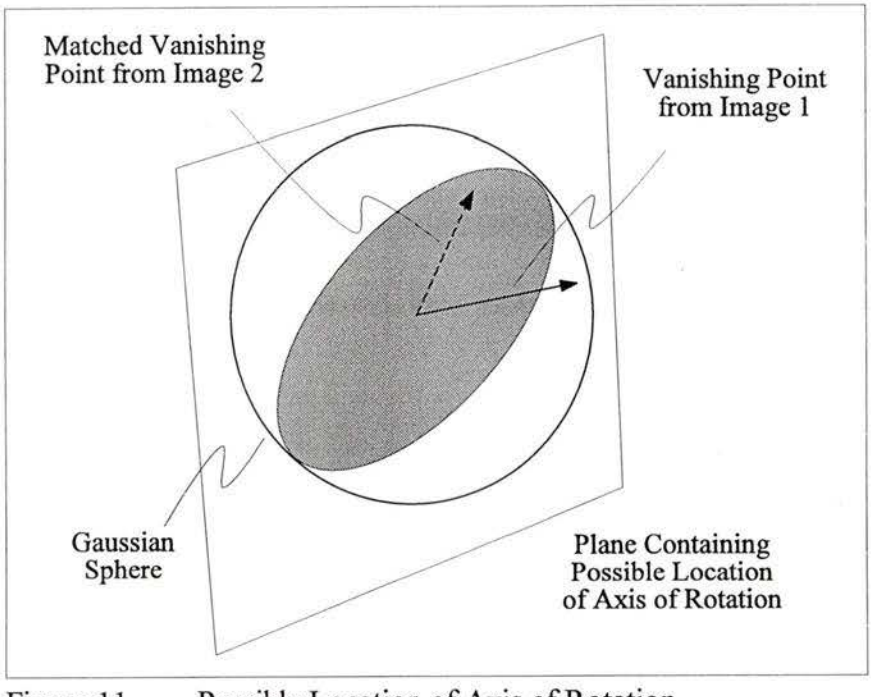


Figure 11 Possible Location of Axis of Rotation

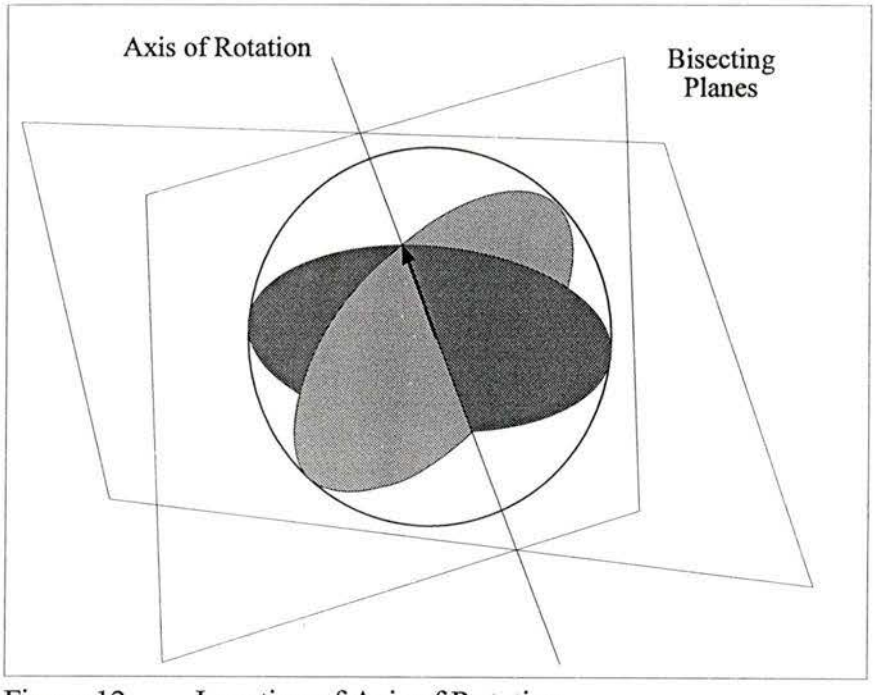


Figure 12 Location of Axis of Rotation

Obtaining the plane equation for each of these bisecting planes requires two vectors on the surface of each plane. The simplest choices are the vector cross product of the matched vanishing points ($\vec{p}_i$) and the vector sum of the two vanishing points ($\vec{q}_i$):

$$\vec{p}_i = \vec{v}_i \times \vec{v}_i'$$
$$\vec{q}_i = \frac{\vec{v}_i + \vec{v}_i'}{| \vec{v}_i + \vec{v}_i' |} \tag{12}$$

where $\vec{v}_i$ is the vanishing point i from first image while $\vec{v}_i'$ is the matching vanishing point i' from the second image. Then the plane equation is:

$$[x\ y\ z]\ \vec{n}_i = 0 \tag{13}$$

where $\vec{n}_i$ is the normal vector of this plane:

$$\vec{n}_i = \vec{p}_i \times \vec{q}_i \tag{14}$$

To locate the axis of rotation, the vector cross product of the two normal vectors from the two bisecting planes can be used:

$$\vec{k} = \vec{n}_i \times \vec{n}_j \tag{15}$$

With the axis of rotation known, the angle of rotation (θ_i) can be calculated by using the dot product rule:

$$\vec{r} \cdot \vec{s} = |\vec{r}|\,|\vec{s}|\cos(\theta_i) \tag{16}$$

where $\vec{r}$ and $\vec{s}$ are projection of vanishing points $\vec{v}_i$ and $\vec{v}_i'$ respectively, onto a plane that is orthogonal to the axis of rotation:

$$\begin{aligned} \vec{r} &= (\vec{k} \times \vec{v}_i) \times \vec{k} \\ \vec{s} &= (\vec{k} \times \vec{v}_i') \times \vec{k} \end{aligned} \tag{17}$$

The cross product between $\vec{r}$ and $\vec{s}$ is examined to determine the sign of this rotation.

$$\theta_i = \begin{cases} \theta_i & \text{if } \vec{r} \times \vec{s} = \vec{k} \\ -\theta_i & \text{if } \vec{r} \times \vec{s} = -\vec{k} \end{cases} \tag{18}$$

Now that the rotation of the vanishing point between images has been determined, the camera rotation is simply the negative θ_i.

Note that for the second matched vanishing point pair j, a second estimate of θ_j will be obtained. For a consistent rotation between the two matched pairs, θ_i and θ_j should be the same.

3.5 Requirements for the Vanishing Point Method

The following requirements are needed in order to use vanishing point matching to recover camera rotations:

1. At least two vanishing point correspondences between two images.
2. An estimate of the five intrinsic camera parameters.
3. Camera rotation can not be too large (i.e. $< 45°$ for rectangular buildings).

At least two vanishing point correspondences are required because a minimum of two vanishing point matches are needed to determine camera rotation. Because of the use of Gaussian sphere representation, an estimate of camera intrinsic parameters is needed. The third requirement is needed because of the theoretical limit of the vanishing point matching methodology. To better understand what is "too large" a rotation, consider a cubic structure with three mutually orthogonal vanishing point vectors. For simplicity, assume that one of these vanishing point vectors is on the axis of rotation. Then, the camera rotation limit can be defined by half the angle between the remaining

two vanishing point vectors. In other words, for a symmetric N sided building, there will be a maximum of N/2 visible horizontal vanishing points in the image and the angle between each vanishing point vectors will be 360/N degrees. For a rotation greater than 180/N degrees, the vanishing points in one image will be incorrectly matched with other neighbouring vanishing points in the second image. Therefore, a general requirement of camera rotation less than 180/N degrees is necessary.

3.6 Summary

In this chapter, the basic properties of vanishing points and their detection have been examined. From the translation invariant property, it is impossible to recover camera translation simply by matching vanishing points between two images. However, due to the rotation invariant property of the vanishing point constellation, not only is a means to properly match the correct vanishing points between two images provided, but a means to filter out the erroneous vanishing point identification is also available. These erroneous identifications generally come from true line intersections in the image, such as a corner of a building. The presence of erroneous identifications requires a matching algorithm that will make use of this invariant constellation property to correctly match the vanishing points. In the next chapter, different techniques to perform this task will be examined.

4. Vanishing Point Matching

In this chapter three different techniques to match vanishing points will be examined. Two of them are based on simple heuristic rules while the third one is based on relational structure matching techniques.

4.1 Heuristic Approaches

Two heuristic methods for matching vanishing points have been examined. These methods rely on the same heuristic rules, but the rules are applied in a different order. Each method and the associated heuristic rules will be discussed in the following sections.

4.1.1 Minimum Arc Distance

The first heuristic method for matching vanishing points is to use minimum arc distance as the primary rule for matching and then use a common rotation criterion for confirmation and to eliminate false matches. Because of the requirement that the camera rotation cannot be "too large", the nearest vanishing point (minimum arc distance) when the two data sets are overlaid onto the same Gaussian sphere should provide the correct match. Also, because there is only one camera rotation, the resulting rotations from all

the matched vanishing points should agree with each other. Therefore, this consistent rotation criterion can be used to eliminate possible false matches. This method is the simplest matching strategy and the easiest to implement. Assuming no noise (natural line intersections being confused as vanishing points) in the data set, this method should also be very effective. However, with noise vanishing points in the data set, there is a large possibility of matching a vanishing point to a noise point simply because the noise point may now be a closer neighbour than the true corresponding vanishing point. Although this type of false match could potentially be removed by the second criterion, it would definitely eliminate a potential good match. Nevertheless, because of the simplicity of this method, it has been implemented and its effectiveness will be further examined with experimental results in the next chapter.

4.1.2 Common Rotation

Another heuristic method is similar to the one above but applies the rules in the reverse order. All possible rotations with all combinations of two matched vanishing point pairs were computed first. Then from this list of possible rotations, the one that has the highest occurrence will be chosen. In the case of a tie, the group that produced global minimum arc distance will depict the correct choice. This approach may allow slightly more tolerance for noise because noise will not provide a good possible rotation cluster. However, as the amount of noise increases, the possibility of the noise forming a good rotation cluster also increases. Again, because of the simplicity of this method, it has been implemented and the results will be discussed in the next chapter.

4.2 Relational Structures

If two sets of features to be matched can be expressed in the form of a relational structure, then this relational information can be used to match the features. Because of the invariant constellation property of vanishing points, a set of true vanishing points will provide an invariant structure under camera rotation while the noise will not. Therefore, the logical step is to match this invariant structure.

Because the vanishing point vector is a point being constrained on a sphere, the relational property of the constellation can be described by using the arc distance or equivalently the angle between the two vanishing point vectors. To represent this structure, each vanishing point becomes a node in a graph. The link between each node represents the relational property (angle, arc distance). This "graph" representation suggested the use of a graph matching technique. The following section will first examine various types of graph matching techniques and their suitability for vanishing point matching.

4.2.1 Graph Matching

Graph matching can generally be separated into two classes: isomorphic and subisomorphic. The general concept of graph isomorphism is that given two graphs, there is an exact one-to-one mapping between each node and each link of the two graphs. However, for subgraph isomorphism, a section of one graph is mapped to a section of

another graph. Normally, isomorphism is too strict a rule for graph matching because it assumes that there will be an exact match between two graphs and no error or noise is allowed. For practical purposes, the general approach for graph matching is subgraph isomorphism where the presence of noise nodes in the graph is allowed [18].

4.2.1.1 Association Graphs

One classical approach to subgraph isomorphism is through the use of an association graph or a match graph [1, 18, 19, 20]. In addition to the two original graphs, another graph called the association graph is constructed where each node represents an assignment between the nodes of the two original graphs to be matched, while the links between two nodes of this association graph exist if the corresponding assignments are compatible.

For example, consider the two graphs $G_A = \{N_A, R_A\}$ and $G_B = \{N_B, R_B\}$ shown in Figure 13 where $N_i = \{n_i^j,\ j = 1 \ldots \text{number of nodes in graph } i\}$ represents the set of all nodes in the graph and $R_i = \{r_i^{jk}\ \forall\ j, k\}$ represents the set of all relations or links between two nodes in the graph. Then the association graph will contain nodes $\{(n_A^j, n_B^k)\ \ \forall\ \ j, k\}$. If the relation $r_A^{jk} = f(n_A^j, n_A^k)$ in graph G_A is compatible with the relation $r_B^{pq} = f(n_B^p, n_B^q)$ in graph G_B, then there exists a link in the association graph between the nodes (n_A^j, n_B^p) and (n_A^k, n_B^q). If the relations are symmetric, then there is also a link between nodes (n_A^j, n_B^q) and (n_A^k, n_B^p).

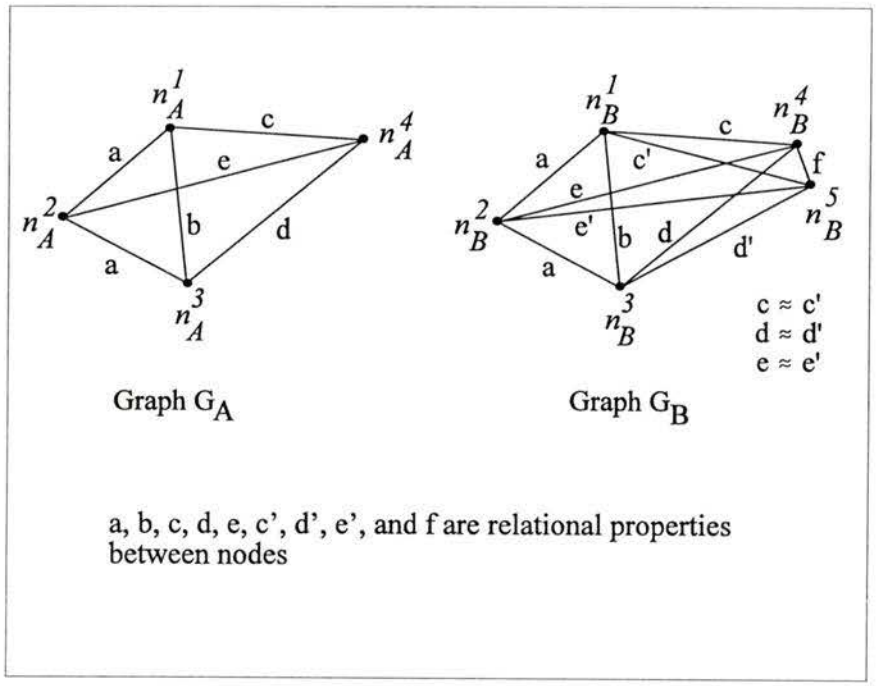


Figure 13 Example Graphs

Table 1 summarizes all the relations from graph G_A and graph G_B. The decision of compatibility between two relations is binary in this example. That is, if two relations are exactly the same or if they are similar[5] (such as c ≈ c'), then the two relations are compatible. Then by comparing the relations between two graphs, the association graph as shown in Figure 14 is constructed. Note that symmetric relations are assumed in this example.

[5]If two relational properties are expressed numerically, then the term similar means that the difference between the two properties are within some specified threshold. For non-numerical properties, such as comparing the positions of two physical objects, then the phrase "an object is on the left side of another object" could be considered to be similar to the phrase "an object is immediately on the left of another object".

Table 1 Example Relations for Graph G_A and Graph G_B

Relation List for Graph G_A	Relation List for Graph G_B
$r_A^{1,2} = f(n_A^1, n_A^2) = a$	$r_B^{1,2} = f(n_B^1, n_B^2) = a$
$r_A^{1,3} = f(n_A^1, n_A^3) = b$	$r_B^{1,3} = f(n_B^1, n_B^3) = b$
$r_A^{1,4} = f(n_A^1, n_A^4) = c$	$r_B^{1,4} = f(n_B^1, n_B^4) = c$
$r_A^{2,3} = f(n_A^2, n_A^3) = a$	$r_B^{1,5} = f(n_B^1, n_B^5) = c'$
$r_A^{2,4} = f(n_A^2, n_A^4) = e$	$r_B^{2,3} = f(n_B^2, n_B^3) = a$
$r_A^{3,4} = f(n_A^3, n_A^4) = d$	$r_B^{2,4} = f(n_B^2, n_B^4) = e$
	$r_B^{2,5} = f(n_B^2, n_B^5) = e'$
	$r_B^{3,4} = f(n_B^3, n_B^4) = d$
	$r_B^{3,5} = f(n_B^3, n_B^5) = d'$
	$r_B^{4,5} = f(n_B^4, n_B^5) = f$

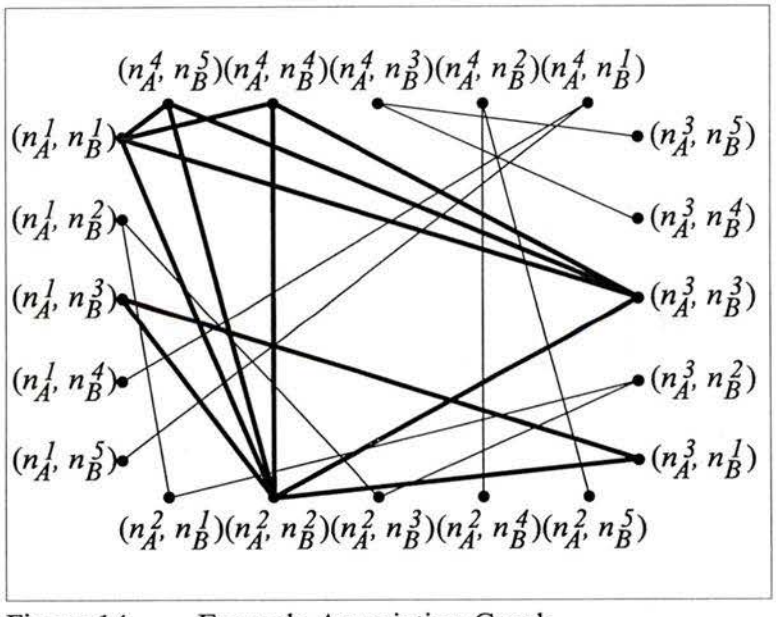


Figure 14 Example Association Graph

Now the main objective is to find the largest maximal cliques[6]. Then the assignments depicted in the nodes of the maximal clique represent the correct match.

The problem with this method is that there is always a good possibility of more than one largest maximal clique, making it difficult to choose the correct match. For instance, in the above association graph example shown in Figure 14, the following cliques are obtained:

Clique 1: (n_A^1, n_B^1), (n_A^2, n_B^2), (n_A^3, n_B^3)

Clique 2: (n_A^1, n_B^1), (n_A^2, n_B^2), (n_A^4, n_B^4)

Clique 3: (n_A^1, n_B^1), (n_A^2, n_B^2), (n_A^4, n_B^5)

Clique 4: (n_A^1, n_B^1), (n_A^3, n_B^3), (n_A^4, n_B^4)

Clique 5: (n_A^1, n_B^1), (n_A^3, n_B^3), (n_A^4, n_B^5)

Clique 6: (n_A^2, n_B^2), (n_A^3, n_B^3), (n_A^4, n_B^4)

Clique 7: (n_A^2, n_B^2), (n_A^3, n_B^3), (n_A^4, n_B^5)

Clique 8: (n_A^1, n_B^1), (n_A^2, n_B^2), (n_A^3, n_B^3), (n_A^4, n_B^4)

Clique 9: (n_A^1, n_B^1), (n_A^2, n_B^2), (n_A^3, n_B^3), (n_A^4, n_B^5)

Clique 10: (n_A^1, n_B^3), (n_A^2, n_B^2), (n_A^3, n_B^1)

Among all these cliques, three of them are maximal cliques: 8, 9, and 10. Note that clique 10 is also a maximal clique because it is not a subset of any other cliques. The two largest maximal cliques (8 and 9) have the size of 4 nodes. By visual inspection, clique 8

[6]A clique is a subgraph such that there is always a link between any given two nodes in the subgraph. A maximal clique is a clique such that it is not being contained in any other cliques.

should be the correct set. However, there are no guidelines to choose which clique is the correct or represents the better set. Therefore, the potential for ambiguity always exists when association graph is used to perform graph matching.

4.2.1.2 Interpretation Trees

Another method commonly used for graph matching is the use of an interpretation tree [1, 21]. Two nodes are arbitrarily matched between the graphs. Then a neighbouring node from each graph is picked and checked to see if they both satisfy a relation constraint. If they do, the process of picking the neighbouring node and attempting to match them is continued until all nodes in one graph are matched to some nodes in the other graph. If they do not, then the steps are backtracked and another neighbouring node is picked until all possible combinations are exhausted. If all of the nodes from one graph cannot be matched to some of the nodes in the other graph, then the path that matches the greatest number of nodes is chosen. This is a depth-first search and backtracking approach. Aside from the time complexity of this approach, it cannot handle noise nodes very well. The possibility of matching noise data from the graph is high because of the stopping criteria that matches all the nodes or as many nodes as possible from one graph to a subgraph.

4.2.1.3 Structural Transformation

Another approach to graph matching is to find the minimum cost of modifying a known model graph into a scene graph [18, 22]. The cost is measured as a function of the

edit operations. These edit operations include inserting nodes, deleting nodes, splitting nodes, merging nodes, inserting links, and deleting links. Some of the practical difficulties with this approach is in deriving an adequate cost function for the edit operations. More importantly, this approach requires the knowledge of a model graph to start with. Since there are no vanishing point model graphs, this approach is not suitable for vanishing point matching.

4.2.1.4 Relaxation Labelling

The last approach to be discussed is the relaxation labelling technique laid out by Hummel and Zucker [23]. Instead of performing a match between two graphs, the problem now turns into a process of assigning a correct label, from a given set of all possible labels, to a node in a given graph. The labelling process is an iterative scheme where the goal is to achieve the most consistent labelling assignment possible by adjusting the assignment weight in each iteration.

To better understand relaxation labelling, consider a graph with a set of labels attached to each node as shown in Figure 15. For simplicity, assume that the set of labels being attached to the nodes is constant. Small letters ($i, j, k, \ldots$) denote nodes in the graph with n nodes, Λ indicates the set of m labels attached to the nodes and the symbol λ denotes individual elements of Λ. Attached to each node i is a weighted labelling assignment indicating the strength of association for a particular label λ to the node. This weighted labelling assignment, denoted by $p_i(\lambda)$, has a value between 0 and 1. An

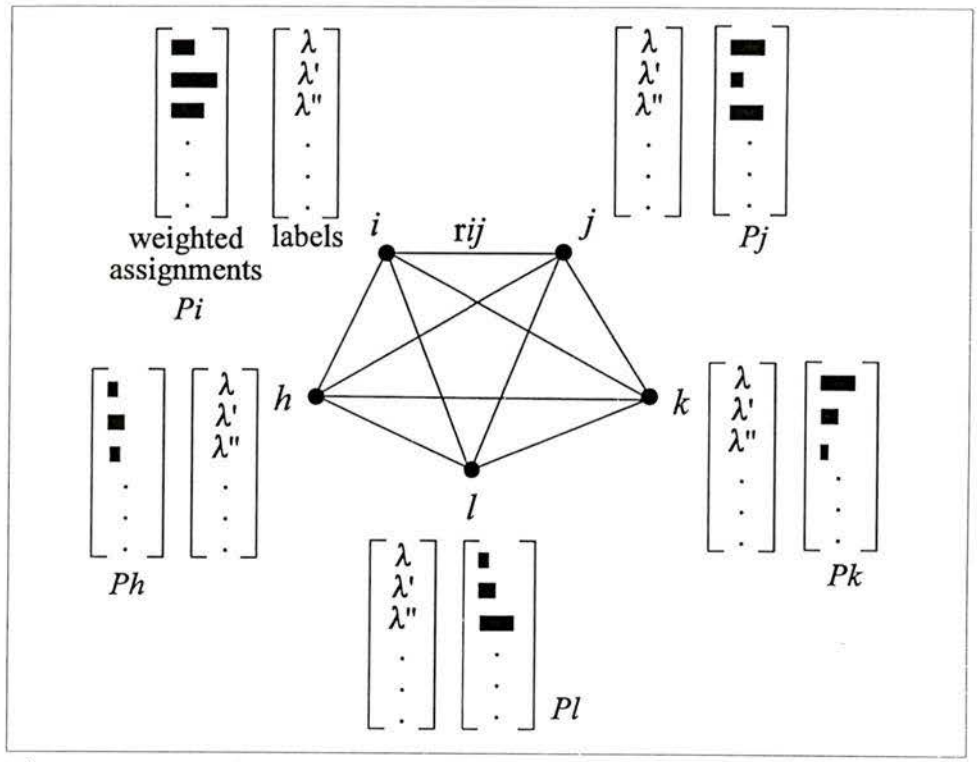


Figure 15 Relaxation Labelling Graph

assignment of 1 means label λ is fully associated with node *i*, while an assignment of 0 means λ is not associated with node *i*. This weighted assignment should also be constrained by:

$$\sum_{\lambda=1}^{m} p_i(\lambda) = 1 \qquad \forall \quad i = 1, \cdots, n \tag{19}$$

A measure of how well label λ is being associated with node *i* while at the same time having label λ' associated with node *j* is called the compatibility function and will be denoted by $r_{ij}(\lambda,\lambda')$. The general behaviour of these compatibility functions should be such that large positive values indicate strong compatibility while large negative values indicate strong incompatibility. Note that relaxation labelling is not restricted only to

relationships between two nodes. If there is a measure for how well label λ is being associated with node *i* while having label λ' associated with node *j* and label λ'' associated with node *k*, then a third-order relationship can be formulated. The compatibility function for this relationship will be denoted by $r_{ijk}(\lambda,\lambda',\lambda'')$.

Relaxation labelling involves an iterative scheme which adjusts the assignment weights until an 'unambiguous' labelling of the nodes has been achieved. Unambiguous assignment means that at a particular node *i*, only one label will have an assignment value of 1 and the rest is 0:

$$p_i(\lambda) = \begin{cases} 1 & \textit{if label } \lambda \textit{ is associated with node } i \\ 0 & \textit{if label } \lambda \textit{ is not associated with node } i \end{cases} \tag{20}$$

This labelling assignment space has been defined by Hummel and Zucker as:

$$\mathrm{K} = \begin{cases} \tilde{p} \in \mathbb{R}^{nm}: & \tilde{p} = [\tilde{p}_1, \cdots, \tilde{p}_n]; \\ & \tilde{p}_i = [p_i(1), \cdots, p_i(m)]^T \in \mathbb{R}^m; \\ & 0 \le p_i(\lambda) \le 1, \quad \forall \ i, \lambda; \\ & \sum_{\lambda=1}^{m} p_i(\lambda) = 1, \quad \forall \ i = 1, \cdots, n \end{cases} \tag{21}$$

The goal now is to establish a consistent iterative scheme which will adjust the labelling assignment such that the process will converge to an unambiguous labelling assignment. Hummel and Zucker defined consistency as:

$$\sum_{\lambda=1}^{m} p_i(\lambda) s_i(\lambda) > \sum_{\lambda=1}^{m} v_i(\lambda) s_i(\lambda), \quad i = 1 \cdots n, \ \tilde{v} \in \mathrm{K}, \ \tilde{v} \neq \tilde{p} \tag{22}$$

where $\tilde{v}$ is any arbitrary weighted labelling assignment and $s_i(\lambda)$ is the support for label λ at node i. For binary relations, this support function is defined as:

$$s_i(\lambda) = \sum_{j=1}^{n} \sum_{\lambda'=1}^{m} r_{ij}(\lambda,\lambda')p_j(\lambda') \tag{23}$$

For higher order relations, this support function becomes:

$$s_i(\lambda) = \sum_{j,\lambda'} \sum_{k,\lambda''} \cdots r_{ijk\ldots}(\lambda,\lambda',\lambda'',\cdots)p_j(\lambda')p_k(\lambda'')\cdots \tag{24}$$

If the assigned label at node i is consistent, then the sum of the weighted support should always be maximum. This measure of consistency provides a guideline to update the labelling assignment. At each iteration, consistency $\sum_{\lambda} p_i(\lambda)s_i(\lambda)$ for each node i should be maximized. Also, because all n local consistencies are increased in each iteration, the total consistency

$$A(\tilde{p}) = \sum_{i} \sum_{\lambda} p_i(\lambda)s_i(\lambda) \tag{25}$$

should also be increased. Although maximizing each individual local consistency is not the same as maximizing the total consistency, Hummel and Zucker have proven that for symmetric relations (that is, $r_{ij} = r_{ji}$), if the total consistency function attains a local maximum at $\tilde{p} \in \mathrm{K}$, then $\tilde{p}$ is a consistent labelling. Therefore, relaxation labelling can be re-formulated as an iterative process that finds the local maxima of $A(\tilde{p})$.

To find the local maxima of $A(\tilde{p})$, Hummel and Zucker used a gradient ascent method [24]. To maximize the increase in $A(\tilde{p})$, the gradient of $A(\tilde{p})$ is used to provide the direction for adjusting the assignment $\tilde{p}$. For second-order relations, this gradient is:

$$\begin{aligned} q_i(\lambda) &= \frac{\partial}{\partial p_i(\lambda)} A(\tilde{p}) \\ &= \frac{\partial}{\partial p_i(\lambda)} \sum r_{ij}(\lambda,\lambda') p_i(\lambda) p_j(\lambda') \\ &= \sum_{j,\lambda'} (r_{ij}(\lambda,\lambda') + r_{ji}(\lambda',\lambda)) p_j(\lambda') \\ &= 2 \sum_{j,\lambda'} r_{ij}(\lambda,\lambda') p_j(\lambda') \\ &= 2\, s_i(\lambda) \end{aligned} \tag{26}$$

While for third order relations, it becomes:

$$\begin{aligned} q_i(\lambda) &= \frac{\partial}{\partial p_i(\lambda)} A(\tilde{p}) \\ &= \frac{\partial}{\partial p_i(\lambda)} \sum r_{ijk}(\lambda,\lambda',\lambda'') p_i(\lambda) p_j(\lambda') p_k(\lambda'') \\ &= \sum_{j,\lambda'} \sum_{k,\lambda''} (r_{ijk}(\lambda,\lambda',\lambda'') + r_{kij}(\lambda'',\lambda,\lambda') + r_{jki}(\lambda',\lambda'',\lambda)) p_j(\lambda') p_k(\lambda'') \\ &= 3 \sum_{j,\lambda'} \sum_{k,\lambda''} r_{ijk}(\lambda,\lambda',\lambda'') p_j(\lambda') p_k(\lambda'') \\ &= 3\, s_i(\lambda) \end{aligned} \tag{27}$$

In general for a k^{th} order relations, the gradient can be computed as:

$$q_i(\lambda) = k\, s_i(\lambda) \tag{28}$$

Then the actual updating direction $\tilde{u}$ for $\tilde{p}$ is the projection of this gradient onto the surface of K such that $\tilde{p} + \tilde{u}$ is in K. The main reason for this gradient projection is because the assignment $\tilde{p}$ is constrained by K. However, the gradient $q_i(\lambda)$ of $A(\tilde{p})$ does not necessary lie on K. Therefore, the direction information of the gradient is extracted by projecting it onto the surface of K.

The algorithm for solving the relaxation labelling problem as developed by Hummel and Zucker [23] can be summarized as follows:

1. Start with an initial labelling assignment: $\tilde{p}^0 \in \mathrm{K}$
2. At each step t, compute the gradient:

$$q_i^t(\lambda) = s_i^t(\lambda) \tag{29}$$

   (Note that the number k in Equation (28) has been dropped because only the direction of the gradient, not the magnitude, is of interest.)
3. Obtain the update direction $\tilde{u}^t$ from the gradient.
4. If $\tilde{u}^t = 0$, stop process.
5. Update the assignment with $\tilde{p}^{t+1} = \tilde{p}^t + h\,\tilde{u}^t$ where h is a step size.
6. Repeat step 2 to 5.

To relate relaxation labelling to a matching problem, each node in the second graph can be treated as a label. It should be noted that among all of the potential graph

matching techniques, this is the only one that can efficiently and practically implement higher order relations (relations between more than two nodes) into the matching process. The construction of the association graph as discussed in Section 4.2.1.1, for example, is already by definition a second-order relation process because the links on the association graph is a relation between two nodes only. In relaxation labelling, however, only the gradient function needs modification to incorporate the higher order relations. This added capability provides an additional assurance of the integrity of the structure to be matched. Most importantly, the relaxation labelling approach allows different possible matches and unambiguous choice of matches with better weighting. Therefore, relaxation labelling will be used to perform vanishing point matching.

4.2.2 Vanishing Point Matching with Relaxation Labelling

To apply the relaxation labelling method to vanishing point matching, nodes of the graph will be treated as vanishing points from the first image while labels will be treated as vanishing points from the second image (nodes of a second graph). The next task is to setup the compatibility function $r_{ij}(\lambda,\lambda')$ or $r_{ijk}(\lambda,\lambda',\lambda'')$ so that the gradient $q_i(\lambda)$ can be computed.

Li's [25] research on invariant matching developed a methodology for matching invariant structures using the relaxation labelling algorithm developed by Hummel and Zucker [23]. Li assumed that most objects can be described by two types of invariant structures: pattern of lines and pattern of points in the image. Li's approach has been

followed because of the similarity between invariant point patterns in the image and invariant vanishing point structures on the Gaussian sphere.

Compatibility is a measure of how well label λ is being associated with node i while at the same time having label λ' being associated with node j. It can also be interpreted as a measure of the difference between the relationship of node i-j pair and the relationship of label λ-λ' pair. Remember that labels are nodes from the second graph, therefore, the label pair λ-λ' do form a relationship.

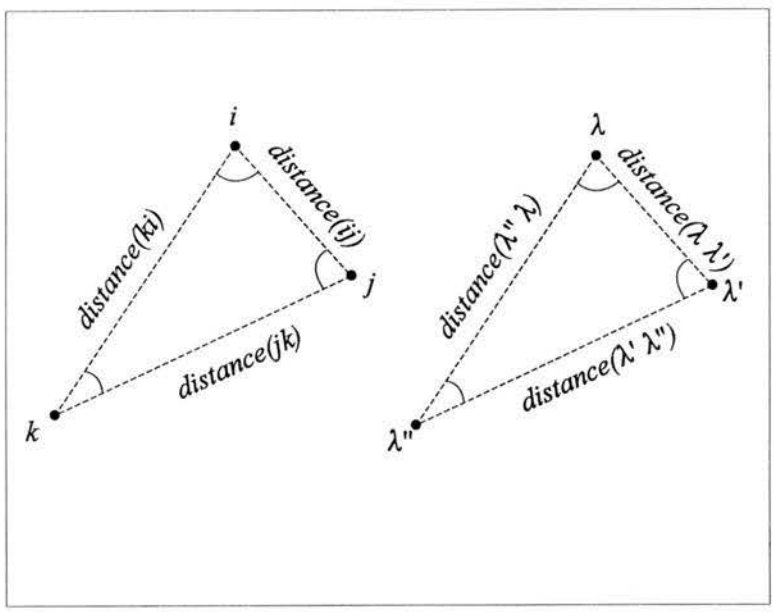


Figure 16 Point Patterns

For the two point patterns shown in Figure 16, two types of relational difference measures are defined:

1. The three interior angles of the triangle formed by three points:

$$Rd_3^{(1)} = | \angle ijk - \angle \lambda\lambda'\lambda'' |$$
$$Rd_3^{(2)} = | \angle jki - \angle \lambda'\lambda''\lambda | \qquad (30)$$
$$Rd_3^{(3)} = | \angle kij - \angle \lambda''\lambda\lambda' |$$

2. The logarithm of the ratio of the triangle's sides:

$$Rd_3^{(4)} = \left| \log\left(\frac{distance(ij)}{distance(jk)} \right) - \left(\frac{distance(\lambda\lambda')}{distance(\lambda'\lambda'')} \right) \right|$$
$$Rd_3^{(5)} = \left| \log\left(\frac{distance(jk)}{distance(ki)} \right) - \left(\frac{distance(\lambda'\lambda'')}{distance(\lambda''\lambda)} \right) \right| \qquad (31)$$
$$Rd_3^{(6)} = \left| \log\left(\frac{distance(ki)}{distance(ij)} \right) - \left(\frac{distance(\lambda''\lambda)}{distance(\lambda\lambda')} \right) \right|$$

The compatibility measure becomes a function of these relational differences. Note that these are all third order relations. The ratio of lengths takes into account of the effects of scale changes, while the use of logarithms provides a better measure of how close the two ratios are to being equal. Also, note that there are a total of 6 different measures for the third-order relational differences.

Recall the definition that these compatibility functions should be set up such that a large positive number should indicate strong compatibility while a large negative number should indicate strong incompatibility. In Li's invariant matching of point patterns (and also in vanishing point matching), the concept of incompatibility between

neighbouring nodes is unclear. Relational differences approaching zero have strong compatibility while larger differences have lower compatibility. Therefore, Li has constrained the compatibility function in the range (0,1]. Also, for each type (α) of relational difference measured with a particular order of relations (β) , the compatibility function should have the form:

$$r_{ijk\cdots}^{(\alpha)}(\lambda,\lambda',\lambda'',\cdots) = g(\rho(Rd_{\beta}^{(\alpha)})) \tag{32}$$

where ρ is a function and should be constrained in the range [0, $+\infty$) while g is another function that maps the previous function ρ into compatibility measure in the range (0,1]. This function g should also have the following three properties:

1. It has the maximum value of 1 at $\rho = 0$.
2. It monotonically decreases as ρ increases.
3. It has the asymptote of 0.

One choice for g is the exponential function

$$r_{ijk\cdots}^{(\alpha)}(\lambda,\lambda',\lambda'',\cdots) = g(\rho(Rd_{\beta}^{(\alpha)})) = \exp(-\rho(Rd_{\beta}^{(\alpha)})) \tag{33}$$

Then combining all difference types (α) of relational differences, the overall compatibility for relation order β becomes:

$$r_{ijk\cdots}(\lambda,\lambda',\lambda'',\cdots) = \prod_{\alpha} r_{ijk\cdots}^{(\alpha)}(\lambda,\lambda',\lambda'',\cdots) = \exp(-\sum_{\alpha} \rho(Rd_{\beta}^{(\alpha)})) \tag{34}$$

In general, the function ρ can be defined as:

$$\rho(Rd_{\beta}^{(\alpha)}) = Rd_{\beta}^{(\alpha)} \; / \; \delta_{\beta}^{(\alpha)} \tag{35}$$

where $\delta_{\beta}^{(\alpha)}$ is some scale factor. Therefore, rewriting Equation (34) gives:

$$r_{ijk\ldots}(\lambda,\lambda',\lambda'',\cdots) = \exp(-\sum_{\alpha} Rd_{\beta}^{(\alpha)} \; / \; \delta_{\beta}^{(\alpha)}) \tag{36}$$

Although there is great similarity between point patterns in the image and vanishing point structures on the Gaussian sphere, they are nevertheless different types of structures and require different relational functions. Since vanishing point vectors are points being constrained onto the surface of the Gaussian sphere, the simplest measure for a second order relationship (R_2) can be defined as the arc distance between the two points or equivalently the angle between the two unit vectors (v_1 and v_2). See Figure 17. Also, because of the use of Gaussian sphere, there are no scale change effects. This means that the ratio of arc distance is not needed as a measure. Therefore, the second-order relationship is:

$$R_2^{(1)}(v_1,v_2) = \cos^{-1}(v_1,v_2) \tag{37}$$

The relational difference, analogous to Equations (30) and (31), is:

$$Rd_2^{(1)} = \mid R_2^{(1)}(i,j) \; - \; R_2^{(1)}(\lambda,\lambda') \mid \tag{38}$$

and the second-order compatibility, Equation (36), is:

$$r_{ij}(\lambda,\lambda') = \exp(-|R_2^{(1)}(i,j) - R_2^{(1)}(\lambda,\lambda')| / \delta_2) \quad (39)$$

where $\beta = 2$, $\alpha=1$ and $\delta_2 = \delta_2^{(1)}$.

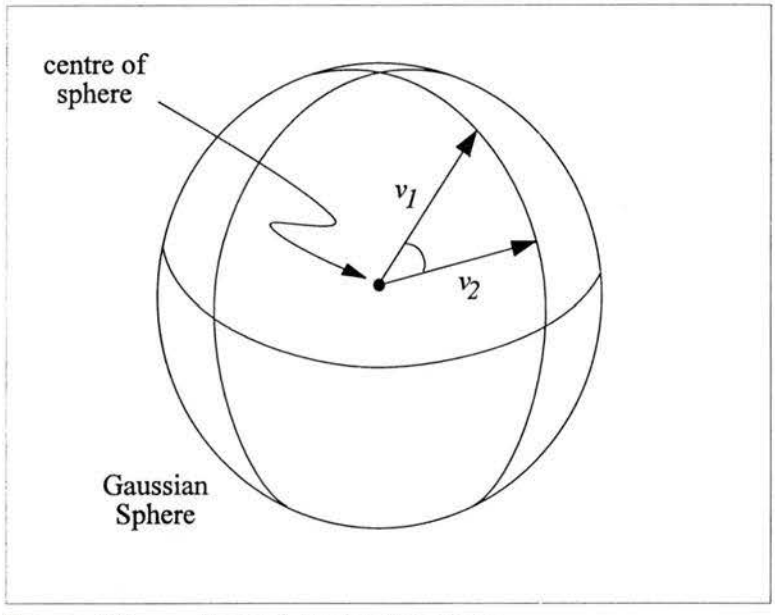


Figure 17 Second Order Relations

For the relationship between three nodes, one natural measure is to use the interior angle between the three planes formed by the three points. See Figure 18. For three vanishing point vectors v_1, v_2, and v_3, this interior angle of v_1 v_2 v_3 can be defined as:

$$\angle v_1 v_2 v_3 = \cos^{-1}(t_1, t_2) \quad (40)$$

where

$$t_1 = \frac{(v_2 \times v_1) \times v_2}{|\,(v_2 \times v_1) \times v_2\,|}$$
$$t_2 = \frac{(v_2 \times v_3) \times v_2}{|\,(v_2 \times v_3) \times v_2\,|} \qquad (41)$$

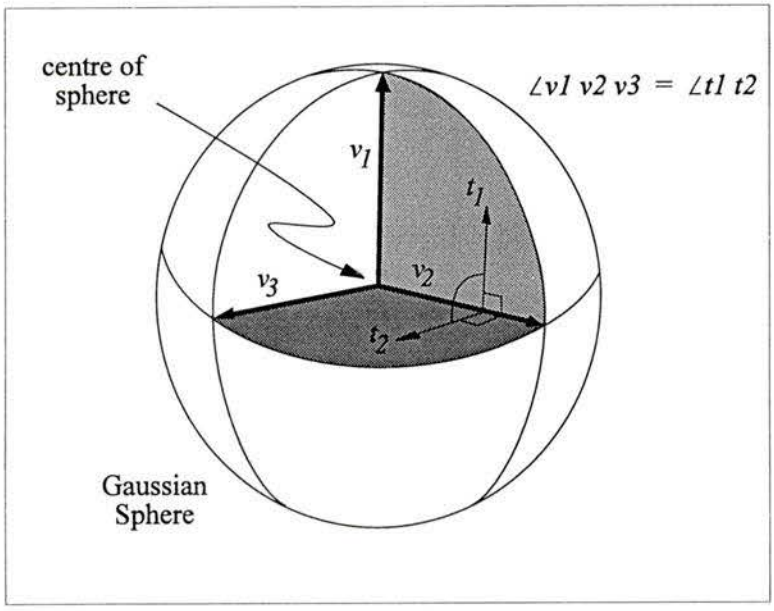


Figure 18 Third Order Relations

Therefore the third-order relationships (R_3) are:

$$R_3^{(1)}(v_1, v_2, v_3) = \angle\, v_1 v_2 v_3$$
$$R_3^{(2)}(v_1, v_2, v_3) = \angle\, v_2 v_3 v_1 \qquad (42)$$
$$R_3^{(3)}(v_1, v_2, v_3) = \angle\, v_3 v_1 v_2$$

Then the relational difference, analogous to Equations (30) and (31), is:

$$\begin{aligned} Rd_3^{(1)} &= | R_3^{(1)}(i,j,k) - R_3^{(1)}(\lambda,\lambda',\lambda'') | = | \angle ijk - \angle \lambda\lambda'\lambda'' | \\ Rd_3^{(2)} &= | R_3^{(2)}(i,j,k) - R_3^{(2)}(\lambda,\lambda',\lambda'') | = | \angle jki - \angle \lambda'\lambda''\lambda | \\ Rd_3^{(3)} &= | R_3^{(3)}(i,j,k) - R_3^{(3)}(\lambda,\lambda',\lambda'') | = | \angle kij - \angle \lambda''\lambda\lambda' | \end{aligned} \tag{43}$$

and the third order compatibility function, Equation (36), is:

$$r_{ijk}(\lambda,\lambda',\lambda'') = \exp\left(- \sum_{\alpha=1}^{3} | R_3^{(\alpha)}(i,j,k) - R_3^{(\alpha)}(\lambda,\lambda',\lambda'') | / \delta_3 \right) \tag{44}$$

where $\beta = 3$ and $\alpha=3$. Here, $\delta_3 = \delta_3^{(1)} = \delta_3^{(2)} = \delta_3^{(3)}$ is assumed because the three-third order relationships came from the same formula.

To combine the effect of both second-order and third-order compatibilities, the gradient equation (29) now becomes a weighted sum of both functions:

$$q_i(\lambda) = \eta_2 \left[\sum_{j,\lambda'} r_{ij}(\lambda,\lambda') p_j(\lambda') \right] + \eta_3 \left[\sum_{j,\lambda',k,\lambda''} r_{ijk}(\lambda,\lambda',\lambda'') p_j(\lambda') p_k(\lambda'') \right] \tag{45}$$

4.2.3 Handling Noise Data in Relaxation Labelling

In vanishing point matching, erroneous vanishing point identifications will be regarded as noise. Ideally, noise data in the first graph should not be assigned any labels. However, because the fundamental theory behind relaxation labelling is to assign a label

λ_i from the set Λ to a node i, it becomes necessary to create an artificial null label ($\lambda = 0$). This null label can be assigned to noise data. Since this null label is artificially created, it has no relationships with other labels and the compatibilities r_{ij} and r_{ijk} are also artificially created whenever a null label is involved:

$$\begin{aligned} r_{ij}(\lambda,\lambda') &= H_2 \qquad \textit{for any} \quad \lambda,\lambda' = 0 \\ r_{ijk}(\lambda,\lambda',\lambda'') &= H_3 \qquad \textit{for any} \quad \lambda,\lambda',\lambda'' = 0 \end{aligned} \tag{46}$$

The choices for H_2 and H_3 are greatly dependent on the set up of the original compatibility functions. Substituting Equation (46) into (39) and (44), produces:

$$\begin{aligned} -\delta_2 \ln(H_2) &= | R_2^{(1)}(i,j) - R_2^{(1)}(\lambda,\lambda') | \qquad \textit{for any} \quad \lambda,\lambda' = 0 \\ -\delta_3 \ln(H_3) &= \sum_{\alpha=1}^{3} | R_3^{(\alpha)}(i,j,k) - R_3^{(\alpha)}(\lambda,\lambda',\lambda'') | \qquad \textit{for any} \quad \lambda,\lambda',\lambda'' = 0 \end{aligned} \tag{47}$$

H_2 and H_3 are effectively thresholds for relational differences. Recall from the previous section that the compatibility function has been interpreted as a measure of how different the relationship of node *i-j* pair and the relationship of label λ-λ' pair are. For a perfect match, the relational differences should be zero and the resulting compatibility r_{ij} will be maximum (1). However, due to effect of image processing, errors in estimation of the vanishing point locations are expected and so the resulting compatibility will be smaller than 1. These H_2 and H_3 values set the limit of how much error is allowed for the relational differences so that a correct match can still be assigned.

4.2.4 Initial Assignment in Relaxation Labelling

The labelling processing has been formulated as an iterative process that finds the

local maxima of the total consistency function $A(\tilde{p})$ (Equation (25)). The term local maxima implies that there is a possibility of having multiple peaks in the total consistency function. Indeed, this possibility does exist in vanishing point matching. Consider an example with three mutually orthogonal vanishing points as shown in Figure 19. There are three matching possibilities but only one of them is the correct match. Therefore, the choice of initial labelling assignment becomes important. Ideally, the initial assignment should be positioned such that it will be on the same "mountain" as the desired local maxima.

One logical method to accomplish this task is to use the same initial criterion as used for the minimum arc distance approach for vanishing point matching (Section 4.1.1). That is, all the initial matches are assigned using the nearest vanishing point from the second image.

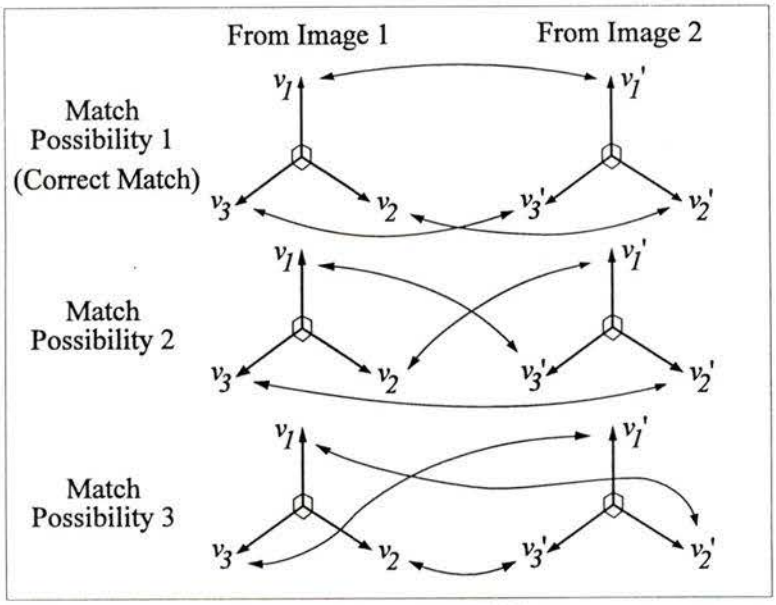


Figure 19 Different Match Possibilities

4.3 Conjugate Vanishing Points

As mentioned in Chapter 3, when the camera is aimed near perpendicular to the parallel line set, the vanishing point location will not be consistent. Although the vanishing point vector is still parallel to the line set, its actual direction may shift from one pole to its conjugate pole (Figure 10). This change in direction will seriously alter the vanishing point structure and affect the vanishing point matching. Figure 20, for example, shows that when one of the vanishing points (v_2) is changed to its conjugate pole (v_2'), the vanishing point structure will be inverted.

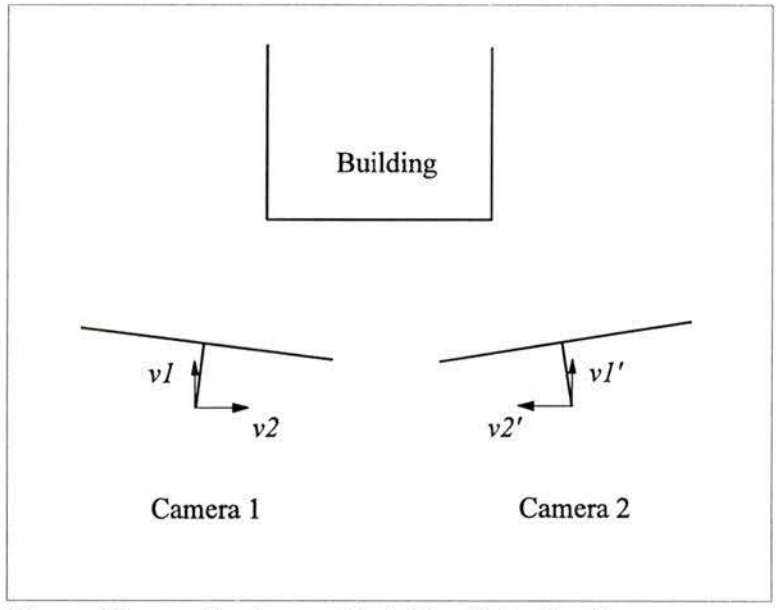


Figure 20 Conjugate Vanishing Point Problem

Ideally, vanishing point v_1 should be matched with vanishing point v_1' and v_2 with v_2'. However, because of this problem of inverted structure, none of the previously described matching approaches can provide satisfactory results.

For the minimum arc distance approach, v_1 will be matched with v_1'. However, v_2 will also be matched with v_1' because it is the nearest neighbour to v_2 when the two cameras overlapped each other (Figure 21). Therefore, the result will be ambiguous matches.

For the common rotation approach, the resulting match will be v_1 to v_2' and v_2 to v_1', because this will be the only possible match that provides a common rotation.

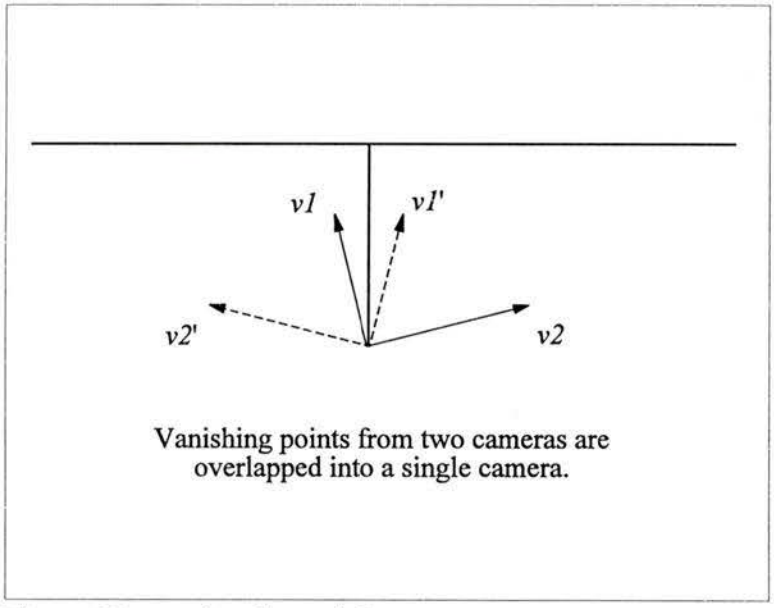


Figure 21 Overlapped Cameras

For relaxation labelling, the same problem persists. Because the structure is inverted, v_1 will be matched to v_2' and v_2 will be matched to v_1'.

However, this problem can be avoided if all the conjugate poles for one set of vanishing points is included in the matching process. Figure 22 and Figure 23 show two possible modified setups.

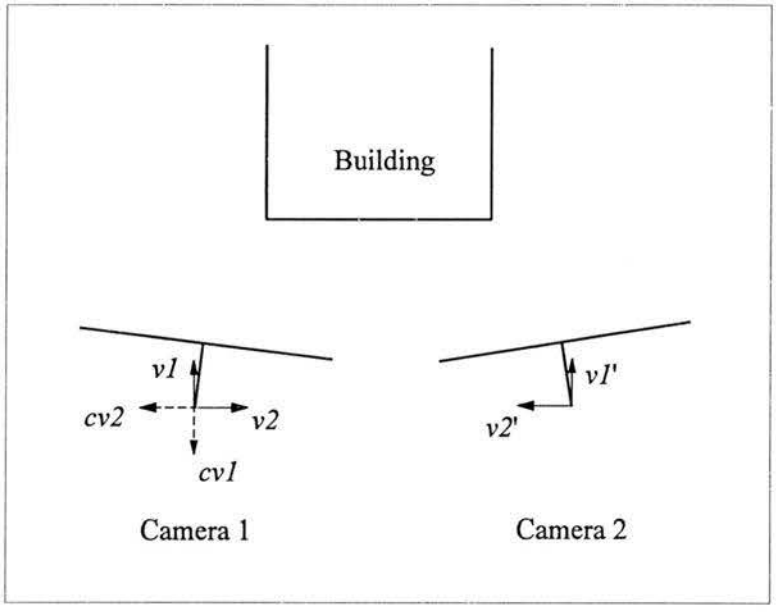


Figure 22 New Camera 1 Vanishing Point Set

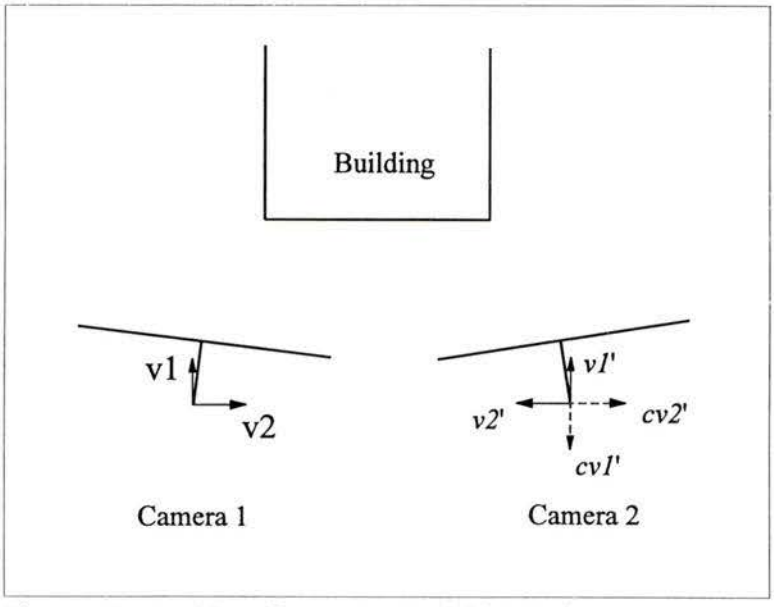


Figure 23 New Camera 2 Vanishing Point Set

In Figure 22, conjugate poles (cv_1 and cv_2) are added into the vanishing point set for camera 1, while in Figure 23, conjugate poles (cv_1' and cv_2') are added into the vanishing point set for camera 2. Using these modified sets, the matching algorithms can match the vanishing points from one image to the conjugate poles from the second image if necessary.

For the minimum arc distance approach, the method shown in Figure 23 is chosen. The closest neighbour for v_2 is now cv_2' which is a correct match. For the common rotation approach, the method shown in Figure 23 has also been chosen. In this case, there are more possible matches that will satisfy the possible common rotation criterion: $\{v_1\text{-}v_1'$ and $v_2\text{-}cv_2'\}$, $\{v_1\text{-}cv_2'$ and $v_2\text{-}cv_1'\}$, $\{v_1\text{-}cv_1'$ and $v_2\text{-}v_2'\}$, and $\{v_1\text{-}v_2'$ and $v_2\text{-}v_1'\}$. However, the group $\{v_1\text{-}v_1'$ and $v_2\text{-}cv_2'\}$ provide the smallest movement (or minimum total arc distances) and will be chosen as the correct matches.

For the relaxation labelling approach, the method shown in Figure 22 is chosen instead of Figure 23 because the latter would simply increase the level of ambiguity for the labelling process. With this new vanishing point set, the sub-structure $\{cv_2$ and $v_1\}$ can be matched with $\{v_2'$ and $v_1'\}$, reinforcing the assignment of label v_1' to v_1. Also, the sub-structure $\{cv_1$ and $v_2\}$ can be matched with $\{v_1'$ and $v_2'\}$. It therefore reinforces the assignment of label v_2' to v_2.

4.4 Summary

In this chapter, four general graph matching techniques for vanishing point matching have been examined: association graphs, interpretation trees, structural transformation, and relaxation labelling. Among the four techniques, relaxation labelling is found to be most suitable for proper vanishing point matching. Therefore, application of relaxation labelling to vanishing point matching has been examined in detail. In addition to the graph matching techniques, two heuristic approaches to vanishing point matching have been presented. The effects of conjugate vanishing points were discussed and methods to solve the problem have been suggested. The next step is to examine the true capability of these algorithms experimentally. In the next chapter, this issue will be addressed and various experiments on the two heuristic approaches and on the relaxation labelling technique will be performed.

5. Vanishing Point Matching Experiments

To examine the performance of the three different approaches to vanishing point matching, four sets of experiments were conducted. Two sets of synthetic data were constructed to examine the theoretical issues of the three different approaches while another two sets of real image data were used to examine the performance in real situations. The remainder of this chapter is divided into four sections. The first two sections will examine these theoretical issues while the last two sections will examine the performance in a real situation using real images.

5.1 Ideal Synthetic Data Study

The purpose of this study is to examine the theoretical issues of the three different types of matching approaches:

1. Capability of the methodologies. Given a perfect data set, it is important to understand whether the three different approaches will function properly and provide correct matches.

2. The amount of noise data (erroneous vanishing point identifications) that the methodologies can handle.

3. How would incorrect estimation of vanishing point location affect the result.

To examine these theoretical issues, an ideal vanishing point structure with known relationships is constructed and rotated a known amount. Noise is then systematically added to the data set in the form of randomly located false vanishing points. To study the sensitivity of the actual estimation of the vanishing point location, this ideal data set is perturbed slightly to simulate incorrect vanishing point position estimation.

5.1.1 Ideal Data Set

Since most buildings possess three mutually orthogonal vanishing points, three vectors that are 90° to each other have been constructed. Two lie near the horizon (the camera's X-Z plane in Figure 3) while the third one lies near the vertical axis. An oblique view of half of the Gaussian sphere that faces the image plane in shown in Figure 24. The vectors were then rotated 25° about the vertical axis $\vec{k} = [\,0\ \ 1\ \ 0\,]^T$ to simulate a camera rotation of -25°. See Figure 25. Note that the camera rotation is directly opposite to the vanishing point rotations.

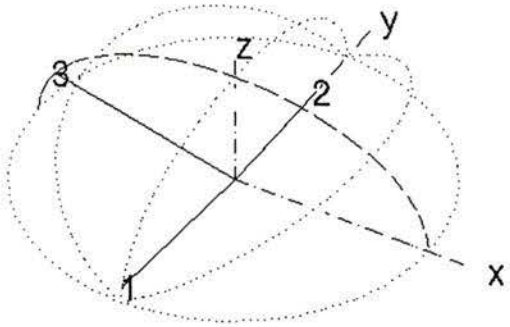


Figure 24 Ideal Vanishing Points

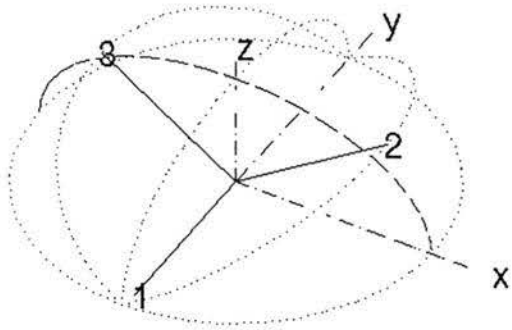


Figure 25 Rotated Vanishing Points

To simulate noise, random data are systematically added to the surface of the Gaussian sphere. These random noise data are Gaussian distributed with zero mean and a standard deviation of 1 radian centred at the Z-axis. Four levels of noise have been examined:

3 noise:	3 vanishing points with 3 random false vanishing points
5 noise:	3 vanishing points with 5 random false vanishing points
7 noise:	3 vanishing points with 7 random false vanishing points
9 noise:	3 vanishing points with 9 random false vanishing points

The first case gives a 1:1 signal to noise ratio while the last case gives a 1:3 signal to noise ratio. Figure 26 to Figure 33 show the vanishing points (VP) with these four levels of noise data on the Gaussian sphere. Table 2 shows the synthetic camera parameters being used in the experiment and Table 3 shows the numerical values of these vanishing point vectors and noise data being projected onto the image plane.

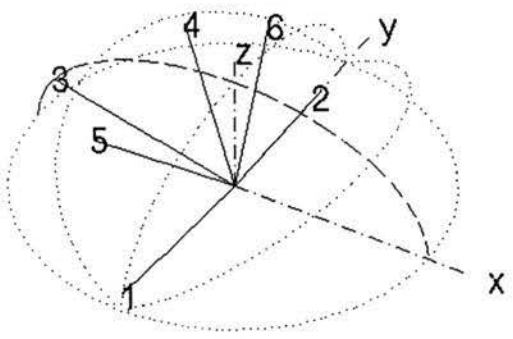


Figure 26 3 Noise, VP Set 1

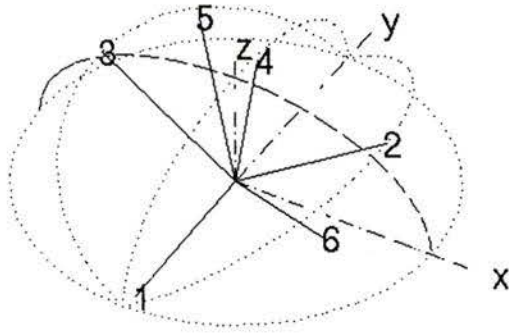


Figure 27 3 Noise, VP Set 2

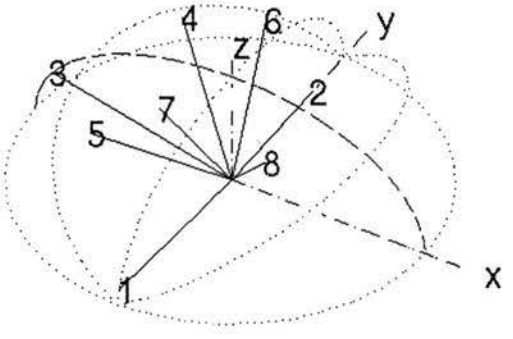


Figure 28 5 Noise, VP Set 1

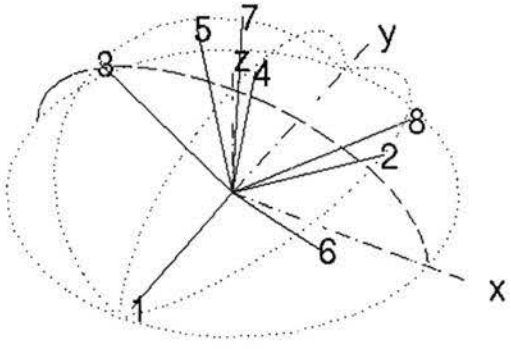


Figure 29 5 Noise, VP Set 2

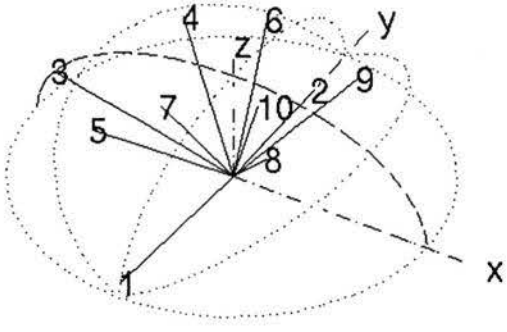


Figure 30 7 Noise, VP Set 1

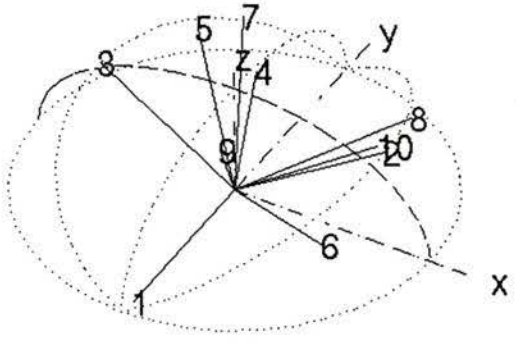


Figure 31 7 Noise, VP Set 2

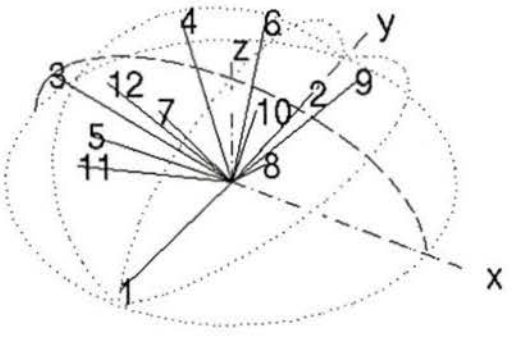


Figure 32 9 Noise, VP Set 1

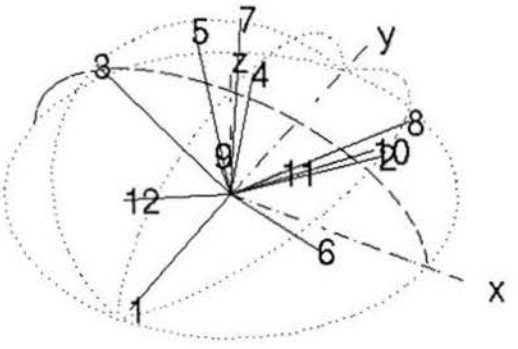


Figure 33 9 Noise, VP Set 2

Table 2 Synthetic Camera Parameters

f = 28 mm	C_x = 381 pixels	C_y = 253 pixels	S_x = 55.555 pix/mm	S_y = 55.555 pix/mm

Table 3 Ideal Vanishing Points and Noise Data Sets

Vanishing point Set 1		Vanishing Point Set 2	
Ideal Synthetic Vanishing Points			
1	(381.00, -10815.24)	1	(1106.36, -11959.45)
2	(952.73, 471.62)	2	(1946.39, 544.11)
3	(-3934.81, 471.62)	3	(-1184.31, 358.16)
Random Noise Data			
4	(-503.57, 956.19)	4	(423.56, 510.10)
5	(-499.67, -785.48)	5	(-403.74, 1021.23)
6	(178.97, 1169.77)	6	(2580.78, -1165.37)
7	(64.29, -268.17)	7	(-443.46, 1895.02)
8	(1065.33, -393.48)	8	(2029.59, 1393.21)
9	(1198.27, 949.40)	9	(625.49, -370.81)
10	(698.13, 51.70)	10	(1776.37, 500.54)
11	(-620.15, -1303.16)	11	(1169.63, -268.81)
12	(-583.55, -118.88)	12	(70.19, -1318.79)

To examine the conjugate vanishing point problem, a similar procedure is used to create the synthetic data set. Figure 34 and Figure 35 show the ideal data set while Figure 36 and Figure 37 show the data set with 1:1 signal to noise ratio. Table 4 shows the numerical values of this data set. The simulated camera rotation about the vertical axis $\vec{k} = [\,0\ \ 1\ \ 0\,]^T$ is -15°. Different noise levels are not examined because the main purpose of this test is to examine the methodology of resolving the conjugate vanishing point problem.

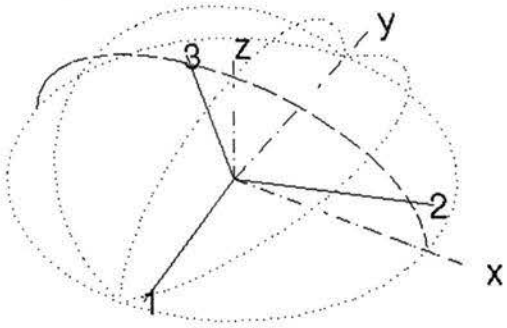


Figure 34 Conjugate VP Set 1

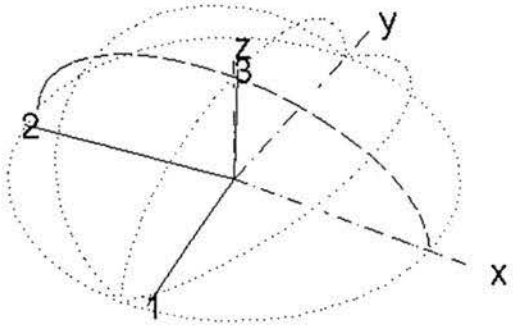


Figure 35 Conjugate VP Set 2

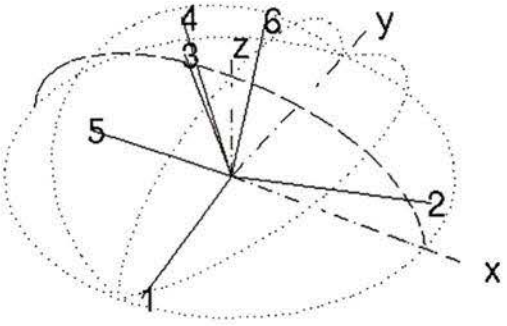


Figure 36 Conj. VP Set 1 with Noise

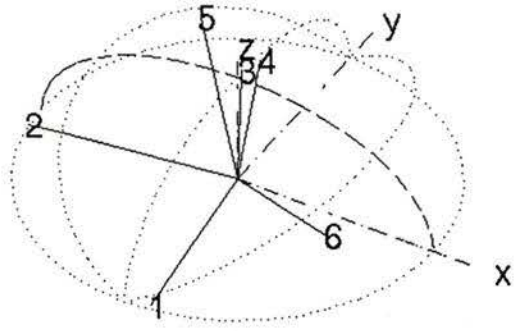


Figure 37 Conj. VP Set 2 with Noise

Table 4 Conjugate Pole Test Data Set

Vanishing Point Set 1		Vanishing Point Set 2	
Ideal Synthetic Vanishing Points			
1	(2602.54, -19043.89)	1	(4654.81, -32108.38)
2	(6260.49, 1055.27)	2	(-492675.20, -64787.79)
3	(-41.03, 329.81)	3	(376.13, 327.13)
Random Noise Data			
4	(-503.57, 956.19)	4	(423.56, 510.10)
5	(-499.67, -785.48)	5	(-403.74, 1021.23)
6	(178.97, 1169.77)	6	(2580.78, -1165.37)

5.1.2 Minimum Arc Distance

The test results for the minimum arc distance approach are summarized in Table 5. For the noise-free case, perfect matches for all three vanishing points are made and the camera rotation is correctly recovered. However, when noise is introduced, one of the good matches (2-2) is eliminated. Examining the data set figures, the closest neighbour for vanishing point #2 from data set 1 is the noise point #4 from data set 2. This 2-4 match does not fit into the secondary common rotation criterion and is therefore eliminated. Although the camera rotations based on the 1-1 and 3-3 matched pairs is still correct, one of the good matches which potentially contained other useful information was lost. For the extremely noisy test case (9 noise point), no match is found. This result shows that this approach cannot handle noise very well. As noise starts to appear, the performance drops unpredictably.

Note that the representation of the camera rotation is not unique. For the noise free case, an axis of rotation of $\vec{k} = [0\ 1\ 0]^T$ with an angle of rotation of -25° was obtained. However, for the cases with noise data, an axis of rotation of $\vec{k} = [0\ -1\ 0]^T$ and an angle of rotation of 25° was computed. These two different representations correspond to the same camera rotations.

The reason for this difference can be explained by the way the camera rotations are computed. Recall from Equation (15) that the axis of rotation is obtained by using the normals of two bisecting planes $\vec{n}_i$ and $\vec{n}_j$:

$$\vec{k} = \vec{n}_i \times \vec{n}_j \tag{48}$$

Reversing the order of $\vec{n}_i$ and $\vec{n}_j$ (due to the matching process) will result with an opposite direction for the axis of rotation:

$$-\vec{k} = \vec{n}_j \times \vec{n}_i = -\vec{n}_i \times \vec{n}_j \tag{49}$$

However, these two vectors do lie on the same axis and therefore represents the same axis of rotation.

For the conjugate vanishing point test case, perfect matches were obtained (see Table 5). This implies that the method of solving the conjugate vanishing point problem functioned as planned.

Table 5 Minimum Arc Distance Match Results for Ideal Synthetic Data

Noise Level Test			
Noise Level	Match Results	Axis of Rotation	Angle of Rotation
0 noise	1-1, 2-2, 3-3	k = [0, 1, 0]	-25°
3 noise	1-1, 3-3	k = [0, -1, 0]	25°
5 noise	1-1, 3-3	k = [0, -1, 0]	25°
7 noise	1-1, 3-3	k = [0, -1, 0]	25°
9 noise	No Match	N/A	N/A
Conjugate Vanishing Point Problem Test			
0 noise	1-1, 2-2, 3-3	k = [0, 1, 0]	-15°
1 noise	1-1, 2-2, 3-3	k = [0, 1, 0]	-15°

5.1.3 Common Rotation

The results for the common rotation approach are summarized in Table 6. This approach managed to obtain a better result for low noise levels. However, for high noise level, only false matches were obtained. Note that ambiguous matches exist when two data points are very close to each other (point #2 and point #10 in data set 2). Also, note that the matching set 1-2, 2-3, and 3-1 formed a valid rotation. Normally, this matching set would be eliminated because it requires a much greater movement than the true set 1-1, 2-2, and 3-3. Due to the addition of noise pairs 1-10, this false matching set obtained higher votes than the true set. As a result, a wrong camera rotation estimate was obtained.

For the conjugate vanishing point test cases, correct results are obtained (see Table 6). This again indicates that the method of solving the conjugate vanishing point problem functioned properly.

Table 6 Common Rotation Match Results for Ideal Synthetic Data

Noise Level Test			
Noise Level	Match Results	Axis of Rotation	Angle of Rotation
0 noise	1-1, 2-2, 3-3	k = [0, 1, 0]	-25°
3 noise	1-1, 2-2, 3-3	k = [0, 1, 0]	-25°
5 noise	1-1, 2-2, 3-3	k = [0, 1, 0]	-25°
7 noise	1-2, 1-10, 2-3, 3-1	k = [-0.2193, 0.5886, 0.7781]	104.63°
9 noise	1-2, 1-10, 2-3, 3-1	k = [-0.2193, 0.5886, 0.7781]	104.63°
Conjugate Vanishing Point Problem Test			
0 noise	1-1, 2-2, 3-3	k = [0, 1, 0]	-15°
1 noise	1-1, 2-2, 3-3	k = [0, 1, 0]	-15°

5.1.4 Relaxation Labelling

The results for the relaxation labelling approach is summarized in Table 7. The relaxation labelling parameters shown in Table 8 produced flawless matches for all tested noise levels. However, the time required for making the matches increases considerably when the noise level increases. Using a SUN SPARC 20 workstation, the test case with 3 vanishing points and 3 noise points took 26 seconds (14 iterations) to complete the task. When the noise level increased to 3 vanishing points and 9 noise points, it took 47 minutes and 18 seconds (19 iterations) to complete. This long completion time is due to

the effect of using both second-order and third-order relations to compute the gradients (Equation (45)). Because the computation for between-node relations is a combinatorial problem, for a graph with n nodes and m labels, there are $\frac{n(n-1)(n-2)}{3!} \cdot m^3$ computations for third-order relations and $\frac{n(n-1)}{2!} \cdot m^2$ computations for second-order relations. Therefore, in each iteration, there are a total of $\frac{n(n-1)(n-2)}{3!} \cdot m^3 + \frac{n(n-1)}{2!} \cdot m^2$ computations to obtain the gradient matrix. However, removing the third order computation and using the second-order relations to compute the gradient still produces correct matches. This requires only 1 second to complete the task for the test case with 3 vanishing points and 3 noise points, and it requires only 27 seconds to complete the task with 9 noise points.

Table 7 Relaxation Labelling Match Results for Ideal Synthetic Data

Noise Level Test			
Noise Level	Match Results	Axis of Rotation	Angle of Rotation
0 noise	1-1, 2-2, 3-3	k = [0, 1, 0]	-25°
3 noise	1-1, 2-2, 3-3	k = [0, 1, 0]	-25°
5 noise	1-1, 2-2, 3-3	k = [0, 1, 0]	-25°
7 noise	1-1, 2-2, 3-3	k = [0, 1, 0]	-25°
9 noise	1-1, 2-2, 3-3	k = [0, 1, 0]	-25°
Conjugate Vanishing Point Problem Test			
0 noise	Process oscillates	N/A	N/A
1 noise	1-1, 2-2, 3-3	k = [0, 1, 0]	-15°

Table 8 Relaxation Labelling Parameters

$H_2 = 0.3$	$\delta_2 = 0.1$	$\eta_2 = 0.5$	$H_3 = 0.01$	$\delta_3 = 0.01$	$\eta_3 = 0.5$

The results for the conjugate vanishing point tests are shown in Table 7. Unlike the two simple approaches, the noise free data set does not converge to a final match using relaxation labelling while the noisy data set does. The reason for this result can be explained by examining the location of the initial labelling assignment on the total consistency function $A(\tilde{p})$ (Equation (25)). Because the initial assignment of 1-1, 2-3, and 3-3 is at the valley (local minima) between the "mountains" of the two local maximas (1-1, 2-2, 3-3) and (1-1, 2-3, 3-2), the process keeps oscillating between the match (1-1, 2-2, 3-3) and (1-1, 2-3, 3-2). However, the noise data actually helps break off this oscillation by providing more matching possibilities instead of the choice of between label 2 or label 3.

5.1.5 Summary

In this study, the performance of three different approaches with the presence of noise data points has been examined. It is found that the two heuristic approaches cannot handle noisy data sets very well. The minimum arc distance approach either loses potential good matches or does not produce any matches at all. The common rotation approach, starts producing false matches when the amount of noise points increases. For

the relaxation labelling approach, correct matches are obtained in all noise level tests. In the study on conjugate vanishing points, it was found that the relaxation labelling method has difficulty converging to a solution for noise free situations due to the problem of improper initial labelling assignment.

5.2 Perturbed Synthetic Data Study

The purpose of this study is to examine a more realistic situation in which the vanishing point locations are not exactly known. The effects on the conjugate vanishing point problem will also be examined.

5.2.1 Perturbed Data Set

To simulate the effect of inaccurate vanishing point locations, the data from Table 3 was perturbed with a Gaussian distribution of 0 mean and a standard deviation of 10 pixels. In this study, the previous four different noise levels are used again. Table 9 to Table 12 show the new perturbed data sets for the four different noise levels.

Table 9 3 Noise, Perturbed Data Set

Vanishing Point Set 1		Vanishing Point Set 2	
Perturbed Synthetic Vanishing Points			
1	(392.45, -10812.56)	1	(1107.56, -11953.59)
2	(954.61, 469.01)	2	(1946.33, 546.97)
3	(-3933.43, 456.08)	3	(-1195.51, 378.58)
Perturbed Noise Data			
4	(-502.17, 951.86)	4	(419.82, 518.64)
5	(-474.17, -792.97)	5	(-409.15, 1011.92)
6	(179.01, 1169.73)	6	(2577.68, -1163.56)

Table 10 5 Noise, Perturbed Data Set

Vanishing Point Set 1		Vanishing Point Set 2	
Perturbed Synthetic Vanishing Points			
1	(383.38, -10822.75)	1	(1102.90, -11959.04)
2	(951.13, 472.22)	2	(1949.64, 559.82)
3	(-3933.19, 471.03)	3	(-1199.44, 351.39)
Perturbed Noise Data			
4	(-512.31, 958.27)	4	(418.35, 513.08)
5	(-499.90, -787.50)	5	(-407.40, 1021.47)
6	(180.79, 1169.10)	6	(2568.72, -1160.03)
7	(70.40, -269.55)	7	(-440.08, 1890.39)
8	(1047.64, -399.34)	8	(2028.03, 1394.38)

Table 11 7 Noise, Perturbed Data Set

Vanishing Point Set 1		Vanishing Point Set 2	
Perturbed Synthetic Vanishing Points			
1	(382.52, -10813.72)	1	(1095.41, -11958.48)
2	(952.81, 470.57)	2	(1963.82, 541.95)
3	(-3939.62, 466.33)	3	(-1185.43, 359.64)
Perturbed Noise Data			
4	(-503.24, 967.25)	4	(418.16, 525.55)
5	(-476.82, -775.09)	5	(-402.74, 1020.18)
6	(186.11, 1165.25)	6	(2581.54, -1162.24)
7	(61.33, -267.41)	7	(-442.37, 1898.74)
8	(1059.27, -385.48)	8	(2033.56, 1403.55)
9	(1192.49, 943.50)	9	(630.78, -368.44)
10	(692.31, 45.52)	10	(1781.12, 506.20)

Table 12 9 Noise, Perturbed Data Set

Vanishing Point Set 1		Vanishing Point Set 2	
Perturbed Synthetic Vanishing Points			
1	(379.51, -10818.22)	1	(1111.94, -11956.60)
2	(949.61, 477.29)	2	(1944.66, 533.84)
3	(-3923.81, 466.51)	3	(-1176.96, 355.01)
Perturbed Noise Data			
4	(-496.46, 957.27)	4	(427.35, 510.56)
5	(-499.19, -784.01)	5	(-412.55, 1020.87)
6	(172.67, 1190.35)	6	(2589.94, -1150.15)
7	(64.04, -268.79)	7	(-435.83, 1909.70)
8	(1064.09, -395.42)	8	(2033.72, 1394.80)
9	(1217.44, 941.92)	9	(631.56, -366.03)
10	(698.87, 53.32)	10	(1771.62, 516.06)
11	(-620.17, -1303.14)	11	(1175.06, -269.00)
12	(-584.19, -116.90)	12	(69.81, -1319.61)

Table 13 shows that this perturbation produces an error of ±2° on the recovered camera rotation if all correct matches were made. Because the four different noise level test cases were perturbed separately, the same set of true vanishing points were perturbed differently and result with slightly different camera rotations in each case.

Table 13 Effects of Perturbed Data on Camera Rotation Recovery

Noise Level	Axis of Rotation	Angle of Rotation
3 Noise	k = [-0.008039, 0.999911, -0.010669]	-25.53°
5 Noise	k = [0.000841, 0.999775, 0.021192]	-23.84°
7 Noise	k = [-0.000206, 0.999988, -0.004822]	-25.26°
9 Noise	k = [0.001077, 0.999789, -0.020497]	-26.45°

For the conjugate vanishing point test, a similar procedure was used to perturb the data set shown in Table 4. Table 14 and Table 15 show the new perturbed data sets.

Table 14 Conjugate Pole Test, Perturbed Data Set

Vanishing Point Set 1		Vanishing Point Set 2	
Perturbed Synthetic Vanishing Points			
1	(2603.67, -19042.31)	1	(4660.20, -32109.19)
2	(6253.55, 1060.51)	2	(-492675.66, -64787.81)
3	(-53.41, 314.36)	6	(374.10, 321.17)

Table 15 Conjugate Pole Test, Perturbed Data Set with Noise Data Point

Vanishing Point Set 1		Vanishing Point Set 2	
Perturbed Synthetic Vanishing Points			
1	(2602.74, -19043.88)	1	(4653.37, -32108.61)
2	(6260.93, 1052.83)	2	(-492673.69, -64788.94)
3	(-49.54, 335.01)	3	(376.96, 322.26)
Perturbed Noise Data			
4	(-505.71, 959.31)	4	(423.57, 510.09)
5	(-495.28, -784.57)	5	(-384.35, 1028.12)
6	(177.00, 1165.92)	6	(2571.58, -1170.18)

5.2.2 Minimum Arc Distance

Table 16 summarizes the results. The results in this test are similar to the one with perfect data set except that no match could be found for the noise level with three noise points. This may appear strange considering the fact that some good matches for the other two noise levels (5 and 7 noise points) were found. The most likely reason for this behaviour is that the perturbation of the data set for the first noise level (3 noise points) is significantly different from the other two noise levels. Therefore, the vanishing point rotation obtained for the pair 1-1 starts to contradict the pair 3-3. Recall from Section 3.4 that after a common axis of rotation is found using two matched pairs (1-1 and 3-3 in this case), the two vanishing point rotations are computed. For valid camera rotations the two rotations should agree. In the implementation of this approach, if the contradiction is greater than 3°, it is decided that no camera rotations can be found. To confirm this hypothesis, the 3° tolerance was relaxed to 7°. The resulting matches are

1-1 and 3-3. The same attempt has also been tried for the case with 9 noise points. However, false matches (3-3, 7-9, and 11-12) are obtained. This indicates that the number of noise points still has a major influence on the matching process even if the tolerance is relaxed.

For the test on conjugate vanishing points, the matching result is the same as the perfect data test cases. This indicates that the solution for solving this problem still functioned properly even if the exact vanishing point locations are not known.

Table 16 Minimum Arc Distance Match Results for Perturbed Synthetic Data

Noise Level Test			
Noise Level	Match Results	Axis of Rotation	Angle of Rotation
3 noise	No Match	N/A	N/A
5 noise	1-1, 3-3	k = [-0.001742, -0.999969, -0.007644]	24.11°
7 noise	1-1, 3-3	k = [0.000201, -0.999988, 0.004912]	25.24°
9 noise	No Match	N/A	N/A
Conjugate Vanishing Point Problem Test			
0 noise	1-1, 2-2, 3-3	k = [-0.017455, 0.999820, -0.007373]	-15.83°
1 noise	1-1, 2-2, 3-3	k = [0.016619, 0.999844, 0.005989]	-14.66°

5.2.3 Common Rotation

The test results, summarized in Table 17, indicate that none of the noise level tests produced correct results. Examining the match results from the 3 and 5 noise points test cases, it appears that there should not be any rotations that can satisfy all three

matches simultaneously. However, recalling how the conjugate vanishing point problem was solved, it becomes clear that the algorithm is matching vanishing point 1 from the first set to the conjugate pole of vanishing point 3 from the second set. Due to the vanishing point perturbations, the computed camera rotations for the matches 1-1, 2-2, and 3-3 no longer agree with each other within a 3° tolerance. Increasing the tolerance to 7° produced proper matches. Unfortunately, this increase in tolerance does not help the cases with higher noise levels (7 and 9 noise points). This result shows that though the conjugate vanishing point problem has been solved, an undesirable artifact has been introduced.

Table 17 Common Rotation Match Results for Perturbed Synthetic Data

Noise Level Test			
Noise Level	Match Results	Axis of Rotation	Angle of Rotation
3 noise	1-3, 2-2, 3-1	k = [-0.538403, 0.086645, -0.838221]	-89.87°
5 noise	1-3, 2-2, 3-1	k = [-0.539137, 0.087254, -0.837686]	-89.31°
7 noise	1-2, 1-10, 2-3, 3-1	k = [-0.218769, 0.587145, 0.779359]	104.65°
9 noise	1-2, 1-10, 2-3, 3-1	k = [-0.220890, 0.588442, 0.777781]	104.67°
Conjugate Vanishing Point Problem Test			
0 noise	1-1, 2-2, 3-3	k = [-0.017455, 0.999820, -0.007373]	-15.83°
1 noise	1-1, 2-2, 3-3	k = [0.016619, 0.999844, 0.005989]	-14.66°

5.2.4 Relaxation Labelling

A summary of the results are provided in Table 18. Similar to the results from the perfect data cases, flawless matches were obtained. Again, the completion time was

higher for the cases with more noise points. The case with 3 noise points took 26 seconds (14 iterations) while the case with 9 noise points took 3 hours 23 minutes and 56 seconds (85 iterations). As the situation becomes less ideal, more time and more iterations will be required for relaxation labelling to converge to a result. However, correct matches can also be achieved using only second-order relations in the labelling process. This requires only 2 seconds for the case with 3 noise points and 33 seconds for the case with 9 noise points.

For the conjugate vanishing point tests, all correct matches were obtained. This suggests that the imperfection of vanishing point location actually helps the relaxation labelling process.

Table 18 Relaxation Labelling Match Results for Perturbed Synthetic Data

Noise Level Test			
Noise Level	Match Results	Axis of Rotation	Angle of Rotation
3 noise	1-1, 2-2, 3-3	k = [-0.008039, 0.999911, -0.010669]	-25.53°
5 noise	1-1, 2-2, 3-3	k = [-0.000841, 0.999775, 0.021192]	-23.84°
7 noise	1-1, 2-2, 3-3	k = [-0.000206, 0.999988, -0.004822]	-25.26°
9 noise	1-1, 2-2, 3-3	k = [0.001077, 0.999789, -0.020497]	-26.45°
Conjugate Vanishing Point Problem Test			
0 noise	1-1, 2-2, 3-3	k = [0.017455, -0.999820, 0.007373]	15.89°
1 noise	1-1, 2-2, 3-3	k = [-0.016619, -0.999844, -0.005989]	14.66°

5.2.5 Summary

In this set of tests, the performance of the three approaches has been examined using uncertain vanishing point locations. The performance of the two heuristic approaches dropped drastically. The minimum arc distance approach exhibits problems in obtaining any matches even at very low noise level (3 noise points) but produces no false matches. The common rotation approach produces only false matches (indicating that it is not likely to function properly in real situations with real images). The relaxation labelling technique produced correct matches for all test cases including the conjugate vanishing point tests.

5.3 Real Image with Known Camera Rotation

The purpose of this test case is to examine how camera rotations can be obtained based on this methodology of using vanishing points with real images. Two simple test images of a cube, where camera rotation is known approximately, will be examined.

5.3.1 Cube Images Data Set

Without precision instruments to measure the camera rotations, an object is rotated while the camera remains stationary. This has the same effect as rotating and translating the camera itself. Figure 38 and Figure 39 show the experimental setup of the camera and a cube. The camera rotation is simulated by rotating the experimental cube

about one corner. The cube's rotation is estimated to be approximately 27°. The camera is mounted on a tripod with a downward tilt of approximately 15° (Figure 39). This means the axis of rotation is approximately at k = [0.0000, -0.9659, -0.2588]. Figure 40 and Figure 41 show the two resulting images.

The camera being used is a Kodak DCS200 digital camera with a lens focal length of 28 mm, a pixel size of 9x9 micrometers, and the resolution of 1524x1012 pixels. To speed up the image processing, the image was reduced to 762x506 pixels. This is equivalent to doubling the pixel size to 18x18 micrometers. Using the image centre as an approximation of the camera's principal centre, the camera intrinsic parameters presented as the synthetic camera parameters, Table 2, were obtained.

To obtain information about the vanishing points in these images, straight lines have to be extracted. Burns' [26] line finding algorithm was used. The algorithm examines the intensity gradient orientation, groups the areas that have a similar gradient orientations, and fits a straight line through these areas. Figure 42 and Figure 43 show the extracted straight lines from the above images. Note that lines from the background have been removed because it would be impossible to rotate the background.

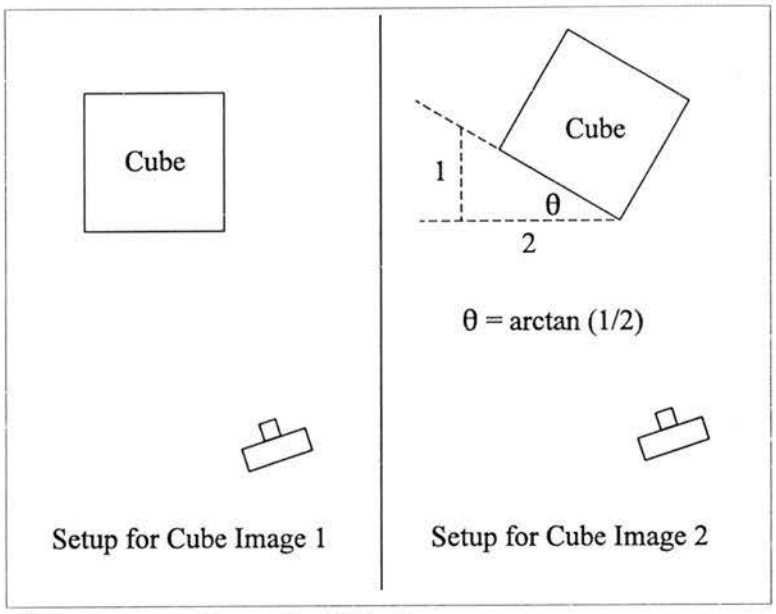


Figure 38 Setup for Cube Images

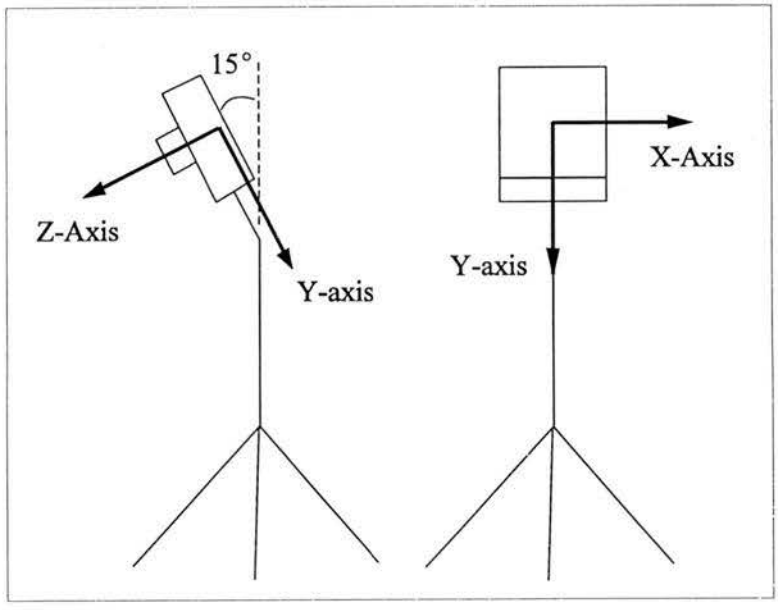


Figure 39 Camera Tilt

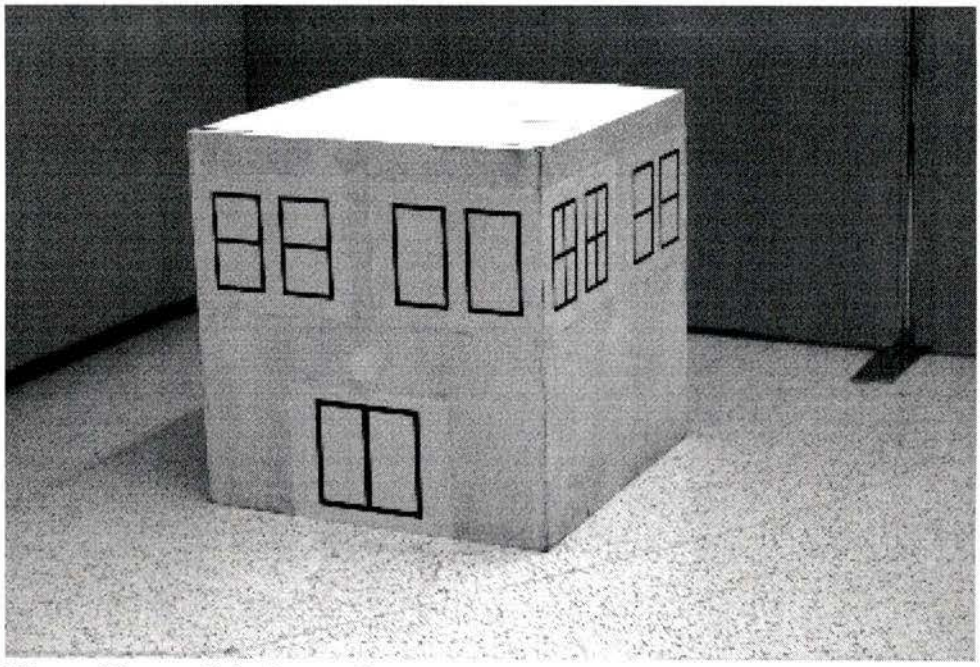


Figure 40 Cube Image 1

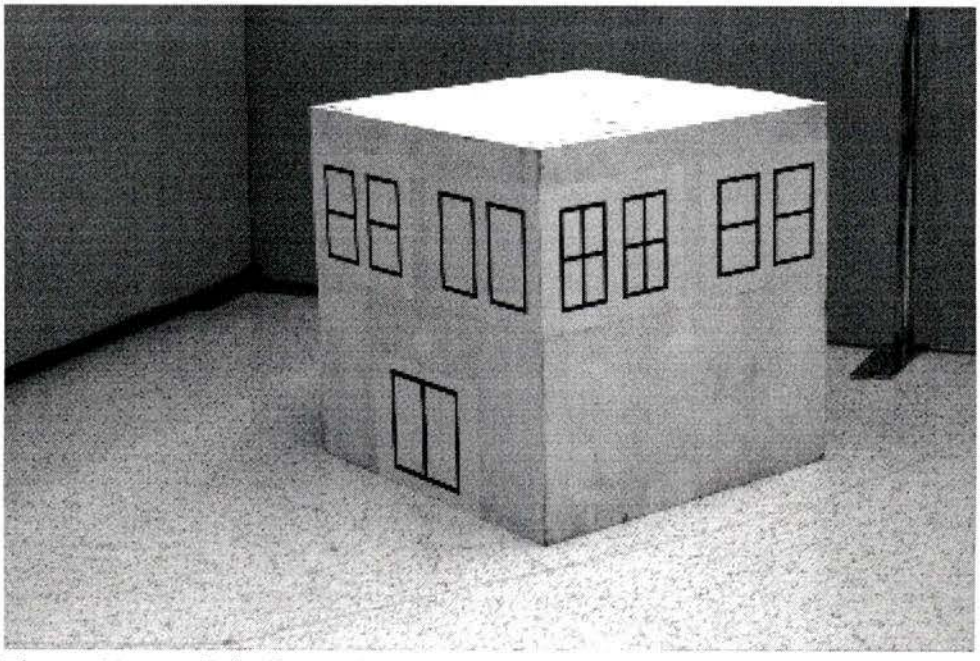


Figure 41 Cube Image 2

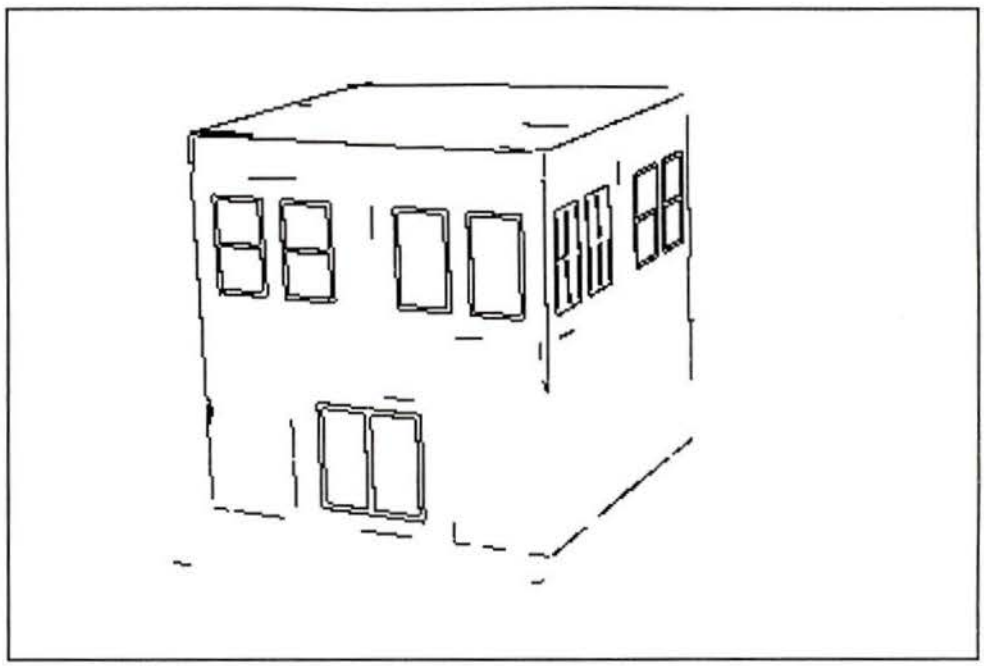


Figure 42 Straight Lines from Cube Image 1

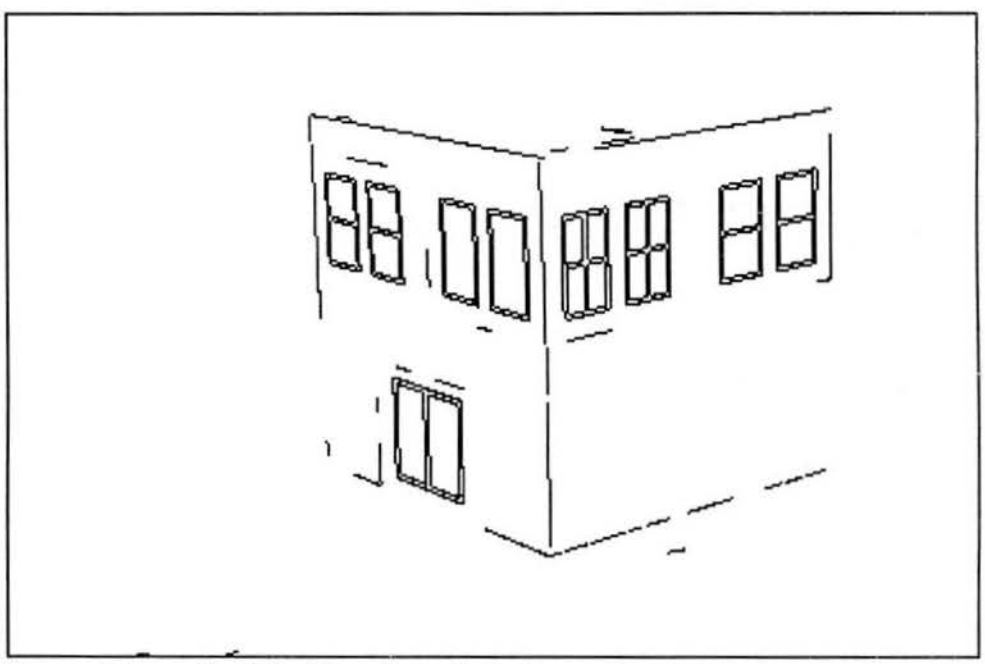


Figure 43 Straight Lines from Cube Image 2

The vanishing points from these images are obtained using an algorithm similar to Magee and Aggarwal's [15] method. All line intersections on the image plane were computed and then converted into the Gaussian sphere representation. Then the vanishing points are assumed to be simply clusters of points on the sphere. The arc distance between two points is used as the measure for this clustering. Figure 44 and Figure 45 show the extracted vanishing points on the Gaussian sphere, with numerical values listed in Table 19. There are noise data in both vanishing point sets. The true vanishing points have been computed by manually extracting two lines on each planar face of the cube and computing their intersection. Then any other lines in the image passing close to this intersection were used to recalculate the true vanishing point (Table 19). Comparing the true vanishing points and the detected vanishing points, there is an error of at least 20 pixels for the coordinates that are near the image area. This means that in this experiment, there is a much higher perturbation of vanishing point location than in the previous study. Results for all vanishing point matching algorithms are shown in Table 20.

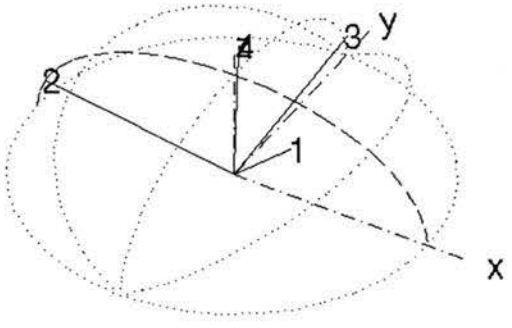


Figure 44 VP from Cube Image 1

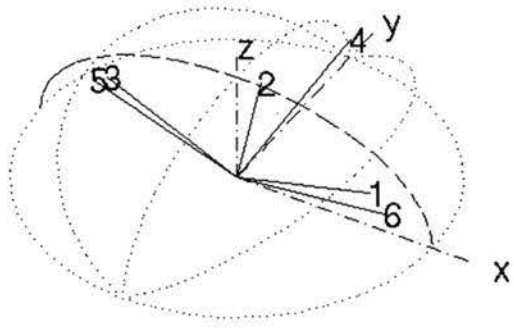


Figure 45 VP from Cube Image 2

Table 19 Cube Images Vanishing Points

Cube Image 1			Cube Image 2		
Detected Vanishing Points					
1	(1175.80, -198.88)	true	1	(2418.89, -207.26)	true
2	(-3067.55, -68.42)	true	2	(562.25, 283.11)	
3	(555.55, 6261.64)	true	3	(-693.80, -72.33)	
4	(233.58, 583.44)		4	(568.04, 6613.45)	true
			5	(-899.81, -131.95)	true
			6	(3305.02, -381.75)	
True Vanishing Points					
1	(1132.65, -178.57)		1	(2786.62, -279.82)	
2	(-2970.58, -63.25)		2	(-870.41, -123.67)	
3	(582.25, 6737.88)		3	(542.00, 6837.50)	

Table 20 Cube Images Match Results

	Match Results	Axis of Rotation	Angle of Rotation
Minimum Arc Dist.	No Match	N/A	N/A
Common Rotation	1-6, 2-3, 3-4	k = [-0.026087, -0.967102, -0.253048]	31.72°
Relaxation Labelling	1-1, 2-5, 3-4	k = [-0.012538, -0.966868, -0.254970]	28.05°
Correct Match	1-1, 2-5, 3-4	k = [-0.012538, -0.966868, -0.254970]	28.05°
Estimated Rotation	N/A	k = [0.000000, -0.965926, -0.258819]	27°

5.3.2 Minimum Arc Distance

The minimum arc distance approach failed to make any matches at all. Examining the vanishing point figures show that the initial minimum distance assignment was 1-1, 2-5, 3-4, and 4-2 (where the first three should be the correct match). However,

similar to the perturbation test, it fails to obtain the final match because of the failure to satisfy a consistent rotation for all matched pairs to within 3° tolerance. This agrees with the previous result that imperfect vanishing point location would greatly degrade the performance of this approach.

5.3.3 Common Rotation

Only one good match (3-4) was obtained and some false matches (1-6 and 2-3) were also produced. Because the false vanishing points 6 and 3 from image 2 are close to the correct vanishing points 1 and 5, respectively, the error from the recovered camera rotations is only 5° and the recovered axis of rotation is similar to the true axis of rotation. However, the false matches could lead to serious problem if knowledge of the lines from the matched vanishing points are required for further image processing.

5.3.4 Relaxation Labelling

Even though the detected vanishing point locations are highly perturbed from their true locations, this approach managed to match the invariant vanishing point structure between the two images. The execution time is only 5 seconds (14 iterations) using both second-order and third-order relations in the labelling process. This fast execution time as compared to the previous studies is due to the fact that there is only one noise point in the first image. Therefore, the number of computations in each iteration is greatly reduced. Again, if only the second-order relation is used in the labelling process, the algorithm will provide correct matches as well. The execution time using only the

second-order relation is 1 second. With perfect matches, the recovered camera rotation is in agreement with the theoretical values to within 1.5°. This result demonstrated the success of using the relaxation labelling approach to match vanishing points between real images. It also shows the success of the technique for recovering camera rotations using matched vanishing points.

5.3.5 Summary

In this experiment, the performance of the three different vanishing point matching approaches using real images was examined. Due to the high perturbation of the detected vanishing point locations from the true vanishing point locations, the two heuristic approaches fail to obtain any satisfactory results. For the relaxation labelling technique, correct matches were obtained. Also, the relaxation labelling result proved that a reasonable estimate of the camera rotation can be recovered successfully when proper vanishing point matches are provided.

5.4 Real Image of a Building

The purpose of this test is to examine a more complicated environment with significantly more noise data than a simple cube image. Therefore, it is important to examine how well the matching algorithms can handle a more complex situation.

5.4.1 Building Images Data Set

Figure 46 and Figure 47 show two images of the same building. Note that there are other objects in the images, such as the car, that will introduce spurious straight lines. Using the Burns [26] line finding algorithm, lines from the two images are extracted and shown in Figure 48 and Figure 49. The extracted line sets show numerous unwanted lines resulting from the trees, car, and shadows. These lines will further complicate the vanishing point detection process. Furthermore, there are more lines from the building itself. The number of true intersections is increased significantly. As a result, there are more noise vanishing points being detected. Table 21 shows the image plane locations of the detected vanishing points while Figure 50 and Figure 51 show the corresponding vanishing point vectors on the Gaussian sphere. True vanishing points were also extracted manually as before and they are shown in Table 21. Results for all matching algorithms are given in Table 22.

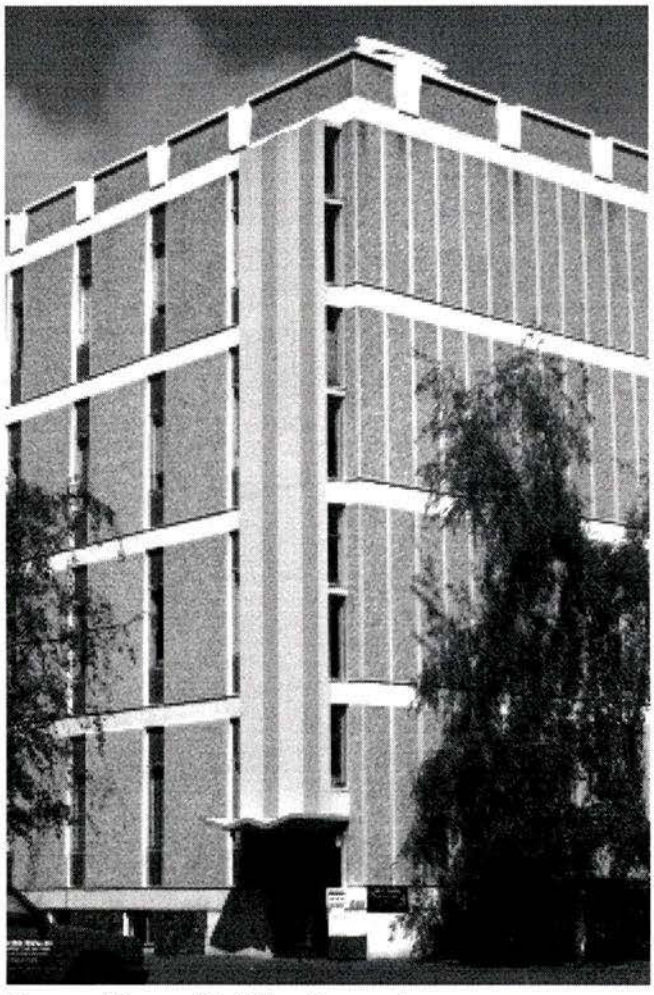


Figure 46 Building Image 1

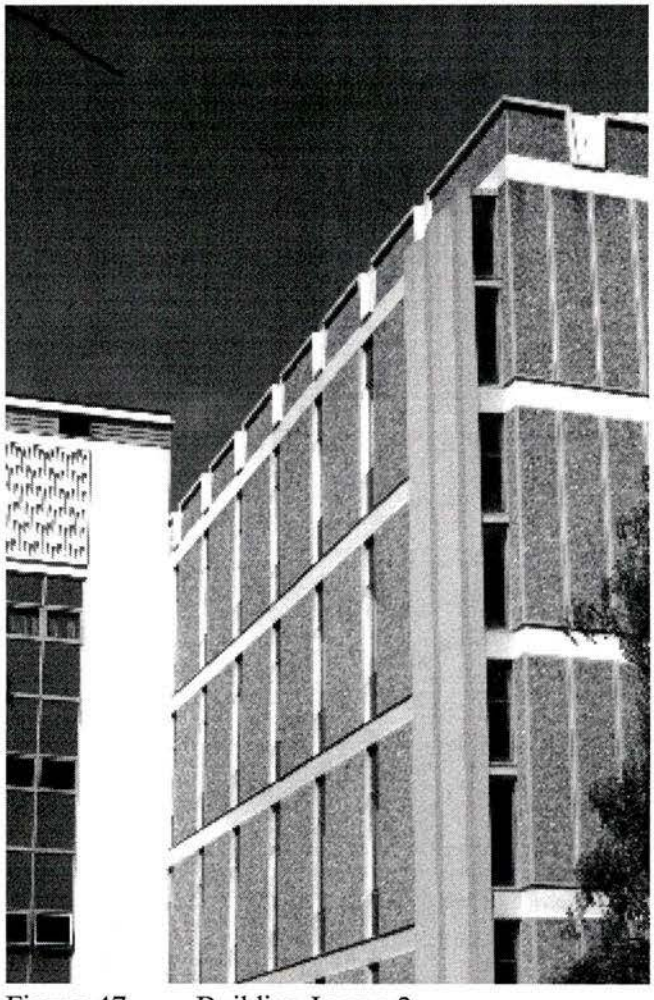


Figure 47 Building Image 2

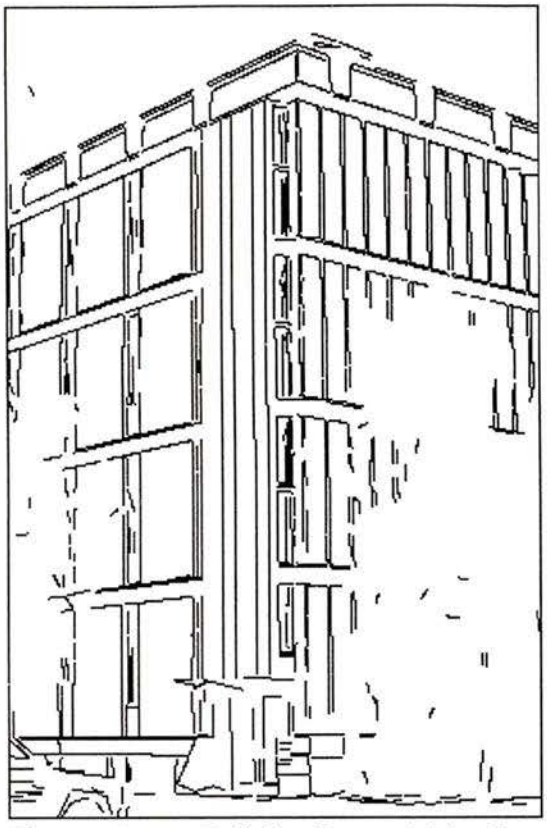


Figure 48 Building Image 1 Line Set

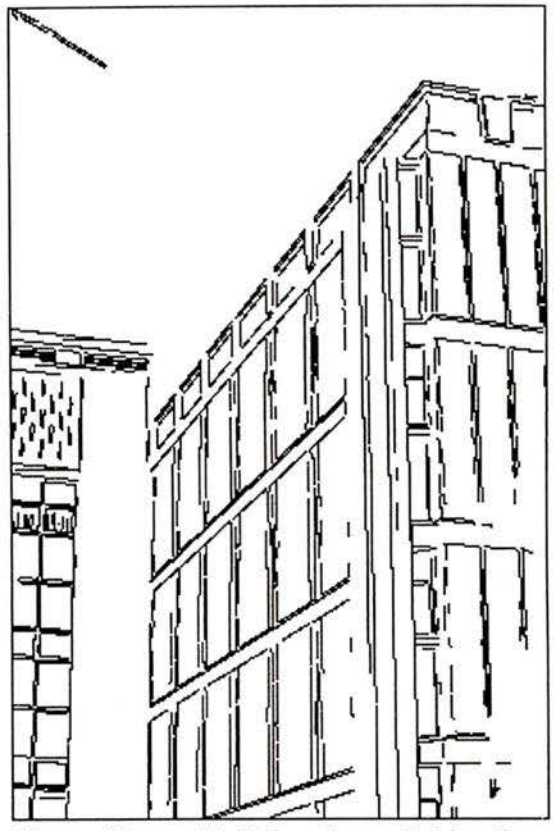


Figure 49 Building Image 2 Line Set

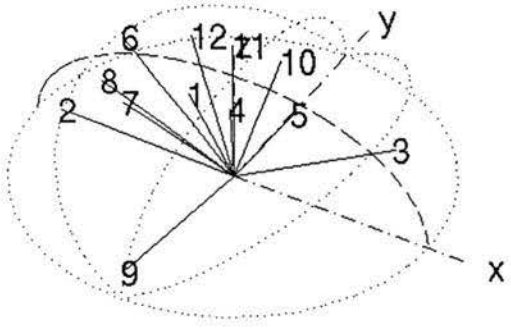


Figure 50 VP from Building Image 1

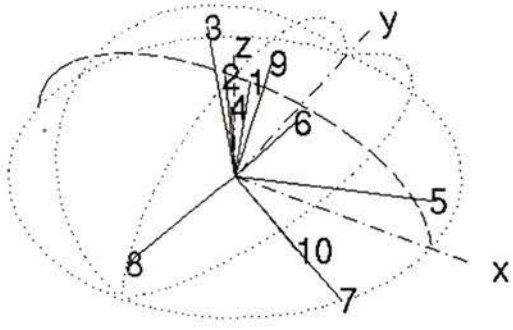


Figure 51 VP from Building Image 2

Table 21 Building Images Vanishing Points

Building Image 1			Building Image 2		
Detected Vanishing Points					
1	(74.34, 60.54)		1	(389.75, 376.44)	
2	(-1341.63, -512.64)		2	(204.52, 353.91)	
3	(2132.79, 683.63)	true	3	(-264.57, 893.38)	true
4	(413.06, 57.04)		4	(408.36, 77.96)	
5	(836.00, 265.21)		5	(5176.25, 919.76)	true
6	(-1128.22, 703.77)	true	6	(885.87, 204.08)	
7	(-418.47, -205.56)		7	(71172.56, -43555.05)	
8	(-832.39, -4.92)		8	(248.48, -4453.98)	true
9	(203.62, -6426.71)	true	9	(403.35, 652.14)	
10	(450.96, 723.28)		10	(2520.97, -1656.86)	
11	(71.82, 689.05)				
12	(-298.60, 705.00)				
True Vanishing Points					
1	(198.25, -6977.70)		1	(222.20, -5460.01)	
2	(-1112.88, 698.01)		2	(-293.15, 922.94)	
3	(2240.91, 712.45)		3	(5479.27, 941.30)	

Table 22 Building Image Match Results

	Match Results	Axis of Rotation	Angle of Rotation
Minimum Arc Dist.	No Match	N/A	N/A
Common Rotation	3-5, 6-8, 9-3, 10-10	k = [-0.84461,-0.31963,-0.42950]	86.54°
Relaxation Labelling	3-5, 6-3, 9-8	k = [0.29069,-0.92994, 0.22519]	21.28°
Correct Match	3-5, 6-3, 9-8	k = [0.29069,-0.92994, 0.22519]	21.28°

5.4.2 Minimum Arc Distance

Similar to the previous cases, this approach did not obtain any matches, confirming that the performance of this approach drops drastically with the presence of noise points and imperfect vanishing point locations.

5.4.3 Common Rotation

Mostly false matches are obtained for the common rotation approach. With the falsely matched vanishing points, the recovered camera rotations is incorrect.

5.4.4 Relaxation Labelling

This technique managed to obtain all correct matches. The estimated camera rotation also appears to be reasonable. One important observation from the successful match result is that the detected false vanishing points have been eliminated at the same time. This means that by successfully matching the vanishing points between two images, true vanishing points and false vanishing points (due to the effect of natural line intersections) can be discriminated.

In the data sets, there are approximately 7 to 9 false vanishing points. Also, the deviation of the detected vanishing point locations from the true vanishing point is approximately 20 pixels. Therefore, this experiment has a much more severe perturbation as compared to the previous study in Section 5.2. The computation time for this experiment is 2 hours 10 minutes and 44 seconds (94 iterations). This execution time

is one hour less than the previous study even though this data set had a much higher perturbation and required more iterations. The main reason for this reduced computation time is that in the previous study, both data sets had 9 noise points while this time only one data set has 9 noise point and the other has only 7 noise points. This demonstrates that the amount of noise data has strong effect on execution time.

An attempt to match the vanishing points with only second-order relations was made and it resulted with the matches: 3-5, 6-3, 9-8, 10-3, 12-9. Although all the correct matches are found, there are also false matches (10-3 and 12-9). This result demonstrated the fact that second-order relations by themselves are not always good enough for robust vanishing point matching (especially when the number of noise data increases and the exact vanishing point locations are not known accurately). However, noticing the fact that all correct matches are obtained with only a few false matches, it suggests that the result is already very close to the desired local maxima. This makes it possible to speed up the third-order process by using the results of the second-order matches as initial assignments.

5.4.5 Summary

In this experiment, the performance of the three matching techniques was examined. The two heuristic approaches exhibit difficulties in providing correct matches. Only the relaxation labelling approach could successfully match the vanishing points between two images. From the successful match results, a discrimination between

true vanishing points and false vanishing points can be made. Because of the amount of noise data and a high level of perturbation of the vanishing point locations, a long execution time resulted. Also, false matches start to exist if only second-order relations were used in the relaxation labelling process. Therefore, a good vanishing point detection algorithm, which can accurately detect vanishing point locations, is still necessary for robust vanishing point matching. An improvement to speed up the relaxation labelling process by using the result of second-order matches as initial assignments has also been suggested.

6. Conclusions and Recommendations

Among various methods of recovering camera rotations, vanishing point matching has been shown to successfully accomplish the task automatically by using only the existing environmental information.

Due to the imperfection of existing vanishing point detection algorithms, natural line intersections can be confused as vanishing points. By using the invariant constellation property from a group of vanishing points, the corresponding vanishing points between two images can be successfully matched. One important finding from this work is that it is now possible to robustly discriminate between true vanishing points and false vanishing points.

Among the three different vanishing point matching approaches examined, relaxation labelling proved to be able to effectively apply the invariant constellation property and successfully match the corresponding vanishing points between two images. However, as the number of false vanishing points in the data sets increased, the computation time also increased. To improve the computation time, a hierarchical relaxation labelling process is recommended by first performing a second-order match, then performing the third-order match using the second-order match results.

Bibliography

1. Faugeras, O. *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Cambridge, Massachusetts, 1993.

2. Spong, M.W. and Vidyasagar, M. *Robot Dynamics and Control*, John Wiley & Sons, New York, 1989.

3. Wolf, P.R. *Elements of Photogrammetry*, 2nd Edition, McGraw-Hill, New York, 1983.

4. Horn, B.K.P. "Relative Orientation", *International Journal of Computer Vision*, Vol. 4, pp. 59-78, 1990.

5. Horn, B.K.P. "Relative Orientation Revisited", *Journal of Optical Society of America A*, Vol. 8, No. 10, pp. 1630-1638, October 1991.

6. Tsai, R. "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, Florida, pp. 364-374, June 1986.

7. Chen, W. and Jiang, B. C. "3-D Camera Calibration Using Vanishing Point Concept", *Pattern Recognition*, Vol. 24, No. 1, pp. 57-67, 1991.

8. Wang, L. and Tsai, W. "Camera Calibration by Vanishing Lines for 3-D Computer Vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 4, pp. 370-376, April 1991.

9. Shigang, L., Tsuji, S., and Imai, M. "Determining of Camera Rotation from Vanishing Points of Lines on Horizontal Planes", *Proceedings of Third International Conference on Computer Vision*, pp. 499-502, 1990.

10. Caprile, B. and Torre, V. "Using Vanishing Points for Camera Calibration", *International Journal of Computer Vision*, Vol. 4, pp. 127-140, 1990.

11. Quan, L. and Mohr, R. "Matching Perspective Images Using Geometric Constraints and Perceptual Grouping", *Proceedings of Second International Conference on Computer Vision*, Tampa, Florida, 1988.

12. Barnard, S.T. "Interpreting Perspective Images", *Artificial Intelligence*, Vol. 21, pp. 435-462, 1983.

13. Quan, L. and Mohr, R. "Determining Perspective Structures Using Hierarchical Hough Transform", *Pattern Recognition Letters*, Vol. 9, No. 4, pp. 279-286, May 1989.

14. Hough, P.V.C. "Method and Means for recognizing complex patterns", U.S. Patent 3,069,654; 1962.

15. Magee, M.J. and Aggarwal, J.K. "Determining Vanishing Points from Perspective Images", *Computer Vision, Graphics, and Image Processing*, Vol. 26, pp. 256-267, 1983.

16. Brillault-O'Mahony, B. "New Method for Vanishing Point Detection", *Computer Vision, Graphics, and Image Processing: Image Understanding*, Vol. 54, No. 2, pp. 289-300, September 1991.

17. McLean, G.F. and Kotturi, D. "Vanishing Point Detection by Line Clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 11, pp. 1090-1095, November 1995.

18. Schalkoff, R.J. *Pattern Recognition: Statistical, Structural and Neural Approaches*, John Wiley & Sons, New York, 1992.

19. Ballard, D.H. and Brown, C.M. *Computer Vision*, Prentice-Hall, New Jersey, 1982.

20. Yang, B., Snyder, W.E., and Bilbro, G.L. "Matching Oversegmented 3D Images to Models Using Association Graphs", *Image and Vision Computing*, Vol. 7, No. 2, pp. 135-143, May 1989.

21. Wong, E.K. "Model Matching in Robot Vision By Subgraph Isomorphism", *Pattern Recognition*, Vol. 25, No. 3, pp. 287-303, 1992.

22. Wang, J.T.L., Zhang, K., and Chirn, G. "The Approximate Graph Matching Problem", *Proceedings of the 12th International Conference on Pattern Recognition*, Vol. II, pp. 284-288, 1994.

23. Hummel, R.A. and Zucker, S.W. "On the Foundation of Relaxation Labelling Processes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No. 3, pp. 267-287, May 1983.

24. Luenberger, D. *Optimization by Vector Space Methods*, John Wiley & Sons, New York, 1969.

25. Li, S.Z. "Matching: Invariant to Translation, Rotation, and Scale Changes", *Pattern Recognition*, Vol. 25, No. 6, pp. 583-594, 1992.

26. Burns, J.B., Hanson, A.R., and Riseman, E.M. "Extracting Straight Lines", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 4, pp. 425-455, July 1986.

Appendix A Geometry of Vanishing Point Vector

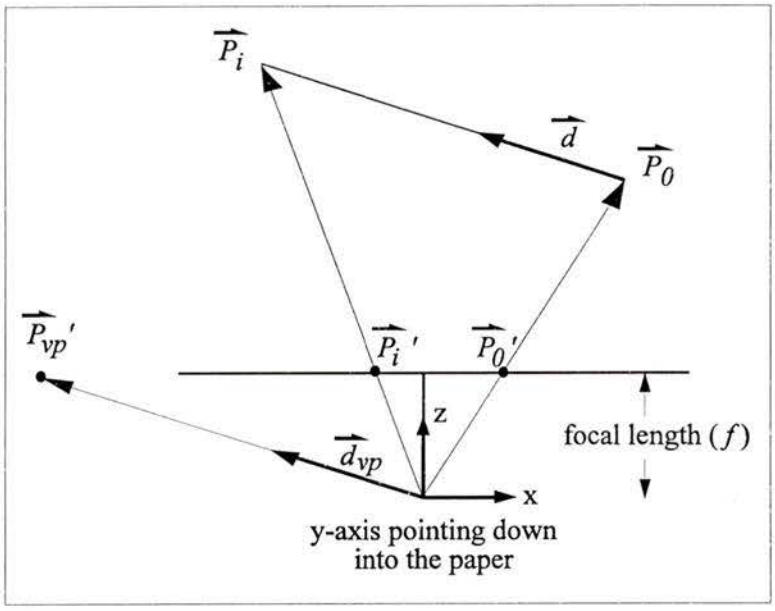


Figure 52 Vanishing Point Vector Geometry

The parametric line equation can be expressed with a point ($\vec{p}_0$) and a direction unit vector ($\vec{d}$) as:

$$\vec{p}_i = \vec{p}_0 + \alpha \vec{d}$$

where α is the length parameter and $\vec{p}_i$ is any point on the line.

Perspective projection of the point $\vec{p}_0$ to the image plane as described by Equation (2) is:

$$\begin{bmatrix} w p_{0xi} \\ w p_{0yi} \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_{0x} \\ p_{0y} \\ p_{0z} \\ 1 \end{bmatrix} = \begin{bmatrix} f p_{0x} \\ f p_{0y} \\ p_{0z} \end{bmatrix}$$

Then,

$$\vec{p}_{0i} = \begin{bmatrix} \frac{f}{p_{0z}} p_{0x} & \frac{f}{p_{0z}} p_{0y} \end{bmatrix}^T$$

Expressing this point in the camera's coordinate system:

$$\vec{p}_0' = \begin{bmatrix} \frac{f}{p_{0z}} p_{0x} & \frac{f}{p_{0z}} p_{0y} & f \end{bmatrix}^T$$

Then for any point on the line:

$$\vec{p}_i' = \begin{bmatrix} \frac{f}{p_{0z} + \alpha d_z} (p_{0x} + \alpha d_x) & \frac{f}{p_{0z} + \alpha d_z} (p_{0y} + \alpha d_y) & f \end{bmatrix}^T$$

where $\vec{d} = [d_x \; d_y \; d_z]$. For vanishing point, $\alpha = \infty$. Then location of the vanishing point on the image plane is:

$$\vec{p}_{vp}' = \lim_{\alpha \to \infty} \begin{bmatrix} \frac{f}{p_{0z} + \alpha d_z} (p_{0x} + \alpha d_x) & \frac{f}{p_{0z} + \alpha d_z} (p_{0x} + \alpha d_x) & f \end{bmatrix}$$

$$= \begin{bmatrix} \frac{f}{d_z} d_x & \frac{f}{d_z} d_y & f \end{bmatrix}$$

The direction unit vector of the vanishing point ($\vec{d}_{vp}$) is:

$$\vec{d}_{vp} = \frac{\vec{p}_{vp}'}{\| \vec{p}_{vp}' \|}$$

where

$$\begin{aligned} \| \vec{p}_{vp}' \| &= \sqrt{\frac{f^2 d_x^2}{d_z^2} + \frac{f^2 d_y^2}{d_z^2} + f^2} \\ &= \frac{f}{d_z} \sqrt{d_x^2 + d_y^2 + d_z^2} \\ &= \frac{f}{d_z} \end{aligned}$$

$$\begin{aligned} \therefore \quad \vec{d}_{vp} &= \frac{d_z}{f} \left[\frac{f}{d_z} d_x \quad \frac{f}{d_z} d_y \quad f \right] \\ &= [\, d_x \quad d_y \quad d_z \,] \\ &= \vec{d} \end{aligned}$$

$\therefore$ The vanishing point vector is parallel to the line in space.

Appendix B Vanishing Point Structure

With two vanishing point unit vectors

$$\vec{d}_1 = [\, d_{1x} \;\; d_{1y} \;\; d_{1z} \,]^T$$

$$\vec{d}_2 = [\, d_{2x} \;\; d_{2y} \;\; d_{2z} \,]^T$$

the angle (or the structure) between them is

$$\theta = \cos^{-1}(\vec{d}_1 \cdot \vec{d}_2)$$

Rotate the vectors by an orthogonal rotation matrix R

$$R = \begin{bmatrix} u_1 & r_1 & s_1 \\ u_2 & r_2 & s_2 \\ u_3 & r_3 & s_3 \end{bmatrix} \quad where \quad \begin{array}{l} \| \vec{u} \| = \| \vec{r} \| = \| \vec{s} \| = 1 \\ \vec{u} \perp \vec{r},\ \vec{r} \perp \vec{s},\ \vec{s} \perp \vec{u} \end{array}$$

Then rotated vanishing point vectors are

$$\vec{d}_1{}' = R\vec{d}_1 = \begin{bmatrix} u_1 d_{1x} + r_1 d_{1y} + s_1 d_{1z} \\ u_2 d_{1x} + r_2 d_{1y} + s_2 d_{1z} \\ u_3 d_{1x} + r_3 d_{1y} + s_3 d_{1z} \end{bmatrix}$$

$$\vec{d}_2{}' = R\vec{d}_2 = \begin{bmatrix} u_1 d_{2x} + r_1 d_{2y} + s_1 d_{2z} \\ u_2 d_{2x} + r_2 d_{2y} + s_2 d_{2z} \\ u_3 d_{2x} + r_3 d_{2y} + s_3 d_{2z} \end{bmatrix}$$

the angle (or the structure) between the two rotated vanishing point vectors are

$$\phi = \cos^{-1}(\vec{d}_1' \cdot \vec{d}_2')$$

$$\begin{aligned}
\vec{d}_1' \cdot \vec{d}_2' &= (u_1^2 + u_2^2 + u_3^2)\, d_{1x} d_{2x} + (u_1 r_1 + u_2 r_2 + u_3 r_3)\, d_{1x} d_{2y} \\
&+ (u_1 s_1 + u_2 s_2 + u_3 s_3)\, d_{1x} d_{2z} + (r_1 u_1 + r_2 u_2 + r_3 u_3)\, d_{1y} d_{2x} \\
&+ (r_1^2 + r_2^2 + r_3^2)\, d_{1y} d_{2y} + (r_1 s_1 + r_2 s_2 + r_3 s_3)\, d_{1y} d_{2z} \\
&+ (s_1 u_1 + s_2 u_2 + s_3 u_3)\, d_{1z} d_{2x} + (s_1 r_1 + s_2 r_2 + s_3 r_3)\, d_{1z} d_{2y} \\
&+ (s_1^2 + s_2^2 + s_3^2)\, d_{1z} d_{2z}
\end{aligned}$$

$$\begin{aligned}
\vec{d}_1' \cdot \vec{d}_2' &= \| \vec{u} \|^2 d_{1x} d_{2x} + (\vec{u} \cdot \vec{r})\, d_{1x} d_{2y} + (\vec{u} \cdot \vec{s})\, d_{1x} d_{2z} \\
&+ (\vec{r} \cdot \vec{u})\, d_{1y} d_{2x} + \| \vec{r} \|^2 d_{1y} d_{2y} + (\vec{r} \cdot \vec{s})\, d_{1y} d_{2z} \\
&+ (\vec{s} \cdot \vec{u})\, d_{1z} d_{2x} + (\vec{s} \cdot \vec{r})\, d_{1z} d_{2y} + \| \vec{s} \|^2 d_{1z} d_{2z}
\end{aligned}$$

$$\because \quad \| \vec{u} \|^2 = \| \vec{r} \|^2 = \| \vec{s} \|^2 = 1$$

and $\quad (\vec{u} \cdot \vec{r}) = (\vec{u} \cdot \vec{s}) = (\vec{r} \cdot \vec{u}) = (\vec{r} \cdot \vec{s}) = (\vec{s} \cdot \vec{u}) = (\vec{s} \cdot \vec{r}) = 0$

$$\begin{aligned}
\therefore \quad \vec{d}_1' \cdot \vec{d}_2' &= d_{1x} d_{2x} + d_{1y} d_{2y} + d_{1z} d_{2z} \\
&= \vec{d}_1 \cdot \vec{d}_2
\end{aligned}$$

$$\begin{aligned}
\therefore \quad \phi &= \cos^{-1}(\vec{d}_1' \cdot \vec{d}_2') \\
&= \cos^{-1}(\vec{d}_1 \cdot \vec{d}_2) \\
&= \theta
\end{aligned}$$

∴ The vanishing point structure is invariant under rotation.

VITA

Surname: Leung Given Names: John Chung Hon

Place of Birth: Hong Kong

Educational Institutions Attended:

University of Waterloo 1988 to 1993

Degrees Awarded:

B.A.Sc. (Honours) University of Waterloo 1993

Publications:

Leung, J.C.H. and McLean, G.F. "Vanishing Point Matching", *Proceedings of IEEE International Conference on Image Processing*, Lausanne, Switzerland, September 16-19, 1996.

Banas, A.O., Carver, M.B., Leung, J.C.H., and Bromley, B.P. "Multidimensional Simulations of Fuel-Rod Appendage Effects on Pressure Drop and Heat Transfer in an Annulus Flow", Presented at the *17th CNS Annual Simulation Symposium*, Kingston, Ontario, August 16-17, 1992.

PARTIAL COPYRIGHT LICENSE

I hereby grant the right to lend my thesis to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Title of Thesis:

Camera Rotation Recovery from Vanishing Point Matching

Author 

John Chung Hon Leung

September 25, 1996.