

Spherical Mosaic Construction using Physical Analogy for Consistent Image Alignment

by

Manuel Guillén González

Thesis submitted to the University of Central Lancashire in
partial fulfilment of the requirements for the degree of

Doctor of Philosophy

October 1999

The work presented in this thesis was carried out in the Department
of Engineering and Product Design at the University of Central Lancashire,
Preston, U.K.

Declaration

I declare that while registered with the University of Central Lancashire for the degree of Doctor of Philosophy I have not been a registered candidate or enrolled student for another award of the University of Central Lancashire or any other academic or professional institution during the research programme. No portion of the work referred to in this thesis has been submitted in support of any application for another degree or qualification of any other University or Institution of learning.

Signed *M Guillen*

To
my parents

and to
Reyes

Acknowledgements

I would like to thank my director of studies Dr. Phil Holifield and my second supervisor Dr. Martin Varley for their encouragement and guidance through all stages of this project.

I am grateful to Professor Trevor J. Terrell for his support and faith in me in the early stages of the project.

I would like to thank Dr. Martyn Shaw for his help with VRML.

The support of the University of Central Lancashire to this research project is gratefully acknowledged, and particularly that of the staff in the Department of Engineering and Product Design.

I thank God for everything He does in my life, particularly for making this work possible.

Abstract

The research contained in this thesis is an investigation into mosaic construction. Mosaic techniques are used to obtain images with a large field of view by assembling a sequence of smaller individual overlapping images.

In existing methods of mosaic construction only successive images are aligned. Accumulation of small alignment errors occur, and in the case of the image path returning to a previous position in the mosaic, a significant mismatch between non-consecutive images will result (looping path problem). A new method for consistently aligning all the images in a mosaic is proposed in this thesis. This is achieved by distribution of the small alignment errors. Each image is allowed to modify its position relative to its neighbour images in the mosaic by a small amount with respect to the computed registration.

Two images recorded by a rotating ideal camera are related by the same transformation that relates the camera's sensor plane at the time the images were captured. When two images overlap, the intensity values in both images coincide through the intersection line of the sensor planes. This intersection line has the property that the images can be seamlessly joined through that line.

An analogy between the images and the physical world is proposed to solve the looping path problem. The images correspond to rigid objects, and these are linked with forces which pull them towards the right positions with respect to their neighbours. That is, every pair of overlapping images are "hinged" through their corresponding intersection line. Aided by another constraint named the spherical constraint, this network of self-organising images has the ability of distributing itself on the surface of a sphere.

As a direct result of the new concepts developed in this research work, spherical mosaics (i.e. mosaics with unlimited horizontal and vertical field of view) can be created.

Contents

Abstract	i
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.2 Aim of the Research	5
1.3 Overview of the Thesis	5
CHAPTER 2 IMAGE ACQUISITION	7
2.1 Introduction	7
2.2 Conventional Camera Technology	7
2.2.1 Image Sensor	7
2.2.2 Field of View and Focal Length	8
2.2.3 Lens Aperture	8
2.2.4 Depth of Field	9
2.2.5 Shutter Speed	10
2.2.6 Video Fields	10
2.2.7 Automatic Gain Control (AGC)	11
2.2.8 Lens Distortion	11
2.2.9 The Ideal Camera	12
2.2.10 The Human Eye	13
2.3 Special Devices for Panoramic Image Acquisition	13
2.3.1 Panoramic Cameras	13
2.3.2 Omnidirectional Cameras	14
2.3.3 Non-frontal Imaging Camera	15
2.4 Summary	16
CHAPTER 3 MOSAIC CONSTRUCTION	17
3.1 Introduction	17
3.2 Image Registration	17
3.2.1 Transformation Models	18

3.2.2 Camera Motion	20
3.2.3 Image Registration Techniques	23
3.2.3.1 Feature Space	23
3.2.3.2 Similarity Measure	23
3.2.3.3 Search Space	24
3.2.3.4 Search Strategy	24
3.2.4 Unwanted Elements	26
3.2.4.1 Motion Parallax	26
3.2.4.2 Lens Distortion	26
3.2.4.3 Moving Objects	27
3.2.4.4 Motion Blur	28
3.3 Image Integration	28
3.3.1 Image Blending	28
3.4 Voronoi Diagrams and Delaunay Triangulation	30
3.5 Summary	34
<u>CHAPTER 4 THE LOOPING PATH PROBLEM AND PROPOSED SOLUTION</u>	35
4.1 Introduction	35
4.2 Previous Approaches	36
4.2.1 Registering New Image with Current Mosaic	37
4.2.2 Registering Sub-mosaics	37
4.2.3 Distribution of the Error	37
4.2.4 Simultaneous Registration of All Images	38
4.3 Proposed Solution to the Looping Path Problem	39
4.4 Minimisation using Physical Simulation	42
4.4.1 Why Physical Simulation?	43
4.4.2 Physical Simulation	44
4.4.2.1 The Rigid Body	44
4.4.2.2 Coordinate Systems	44
4.4.2.3 Linear Equations of Motion	46
4.4.2.4 Angular Equations of Motion	47
4.4.2.5 Simulation Engine	48
4.4.2.6 Forces	49
4.4.2.7 Euler's Integration Method	49

4.4.2.8 Midpoint and Runge-Kutta Integration Methods	52
4.4.3 Physical Model for the Images	54
4.4.4 The Hinge Constraint	55
4.4.5 Stability and Convergence of Image System	58
4.4.5.1 Spring-Mass System	58
4.4.5.2 Damped Spring-Mass System	60
4.4.5.3 Image System	65
4.4.5.4 Example	68
4.5 Summary	70
CHAPTER 5 PLANAR SCENE MOSAIC	71
5.1 Introduction	71
5.2 Image Acquisition	71
5.3 Implementation of Image Registration	73
5.3.1 Transformation Model	73
5.3.2 Feature Space	74
5.3.3 Similarity Measure	75
5.3.4 Search Space	77
5.3.5 Search Strategy	77
5.3.5.1 Hierarchical Pyramids	78
5.3.5.2 Rotation	80
5.3.5.3 Image Registration with Progressive Complexity Models	81
5.3.5.4 Subpixel Accuracy	84
5.4 Physical Model for the Planar Mosaic	85
5.5 Image Integration	86
5.6 Experimental Results on Document Mosaic	86
5.7 Summary	93
CHAPTER 6 SPHERICAL MOSAIC	94
6.1 Introduction	94
6.2 Image Registration Considerations	94
6.3 Partial Spherical Mosaic Mapped to a Plane	95
6.3.1 Image Acquisition	95
6.3.2 Physical Model for Image Alignment	96

6.3.3 Image Integration	97
6.3.4 Results	97
6.3.5 Summary	99
6.4 Double Band Panoramic Mosaic	99
6.4.1 Image Acquisition	100
6.4.2 Physical Model for Image Alignment	101
6.4.2.1 The Spherical Constraint	103
6.4.2.2 The Same Plane Constraint	107
6.4.2.3 Hinge Parameters	108
6.4.2.4 Effects of Damping on Convergence	109
6.4.2.5 Voronoi Tessellation and Delaunay Diagram on the Sphere	110
6.4.3 Image Integration	110
6.4.4 Results	111
6.5 Full Spherical Mosaic	114
6.5.1 Image Acquisition	114
6.5.2 Physical Model for Image Alignment	115
6.5.3 Image Integration and Display	120
6.5.4 Displaying Results	121
6.6 Summary	122
<u>CHAPTER 7 CONCLUSIONS & FURTHER WORK</u>	127
7.1 Contributions	127
7.2 Further Work	129
7.3 Summary	131
<u>REFERENCES</u>	132
<u>APPENDIX A CD-ROM</u>	139
<u>APPENDIX B PUBLICATIONS</u>	141

Chapter 1 INTRODUCTION

1.1 Background

Mosaics are images with a large field of view obtained by assembling two or more individual overlapping images. Historically, the most significant application for image mosaicing is in the development of aerial images [Wolf, 1974] and satellite images [Horii, Oshima and Hirao, 1995], [Milgram, 1975]. These mosaics are commonly used as substitutes for maps and also for remote sensing applications. With the arrival of digital photography, new applications for mosaicing were created [Milgram, 1975] [Milgram, 1977] [Peleg, 1981] [Burt and Adelson, 1983a]. The list of applications using mosaic techniques is extensive today, a typical application being the creation of panoramas (images with a 360° horizontal field of view).

Mosaics can be useful where the image sensor does not have sufficient pixel resolution to provide a single image with the level of detail needed. The attainable resolution in a mosaic is essentially unlimited [Szeliski, 1994a]. Since the images can be acquired using any optical technology (from microscopy, through hand-held video cameras, to satellite photography), the range or scale of the scene being reconstructed is not an issue. Analogously, the field of view is not a limiting factor in a mosaic. Panoramas have a 360° horizontal field of view, and full spherical mosaics with an unlimited vertical field of view are also possible.

The use of video as a source of images for constructing mosaics has been recently investigated by many researchers [Giaccone, Greenhill and Jones, 1998], [Burt, Hansen and Anandan, 1996], [Irani, Anandan and Hsu, 1995], [Mann and Picard, 1995], [Peleg and Herman, 1997b], [Rousso, Peleg and Finci, 1997b], [Bender, 1993], [Szeliski, 1994b], etc. Video sequences lend themselves apropos to mosaicing for a number of

reasons. Considering the high frame-rate of video, typically 25 frames per second, copious amounts of images are available, and the overlapping –indispensable to align the images– between consecutive frames is intrinsically large. In addition, mosaics can achieve image sizes far beyond the resolution of individual video images, and detail can be maintained over an unlimited field of view, therefore enhancing the quality of the images.

Mosaics have a great deal to offer in virtual reality applications. In a virtual reality experience, the main subjective measure of image quality used in the assessment of geometric rendering systems is the degree with which the resulting images are indistinguishable from photographs. One of the biggest bottlenecks standing in the way of widespread virtual reality applications is the slow and tedious model-building process if the models are to resemble the complex and detailed reality with a degree of fidelity. Image-based rendering is an efficient way to create a virtual reality experience. Instead of building and rendering a complete 3D model of the environment, i.e. using polygons, a collection of images is used to render the scene while supporting virtual camera motion. The progression toward image-based rendering systems was initially motivated by the desire to increase the visual realism of the approximate geometric descriptions by mapping images onto their surface (texture mapping) [Heckbert, 1989]. Next, the images were used to approximate global illumination effects (environment map, that is, a 360° spherical image of the environment, to correctly generate reflections from shiny objects [Greene, 1986]), and, most recently, systems have been developed where the images themselves constitute the significant aspects of the scene's description [Chen and Williams, 1993] [Smooth Move].

The class of virtual reality applications which attempt to recreate true reality as convincingly as possible commonly use an environment map which serves as a backdrop and to correctly generate reflections from shiny objects [Greene, 1986]. Examples of these applications include flight simulators, which were among the earliest uses of virtual reality, computer games, medical simulators, visualisation tools, interactive 3D walkthroughs of existing buildings for re-modelling or selling [Szeliski, 1994a]. Walkthroughs of historic building (e.g. palaces or museums) can be used for educational and entertainment purposes. In addition, a museum scenario might include the ability to

look at individual 3D objects such as sculptures, and to bring up related information in a hypertext system [Miller et al., 1991]. Panoramic images, also *called immersive photographs*, are already being used by property and travel companies and museums, to give virtual tours on computers or kiosks [Eccles, 1999]. Other obvious customers for this new technology are hotels, theme parks, estate agents, film and television location finding services, and developers of game and education software. It enables them to obtain images from scenes that are not practical to render artificially.

The film and video industry may also benefit from mosaicing. Many film and video sequences are captured by cameras undergoing small rotational and zoom velocities with near zero translation. Under such conditions the projected view volume is free from perspective changes such as occlusions and parallax, and the image sequence may be merged to create a large virtual image [Giaccone, Greenhill and Jones, 1998]. Since successive images within a video sequence usually overlap by a large amount, the mosaic image provides a significant reduction in the total amount of data needed to represent the scene (i.e. video compression). Moreover, mosaics have been recognised as efficient ways of providing complete representation of large video sequences for film and video editing [Irani, Anandan and Hsu, 1995].

Mosaicing makes possible a new generation of tools for the post-production industry. Mosaicing techniques offer the post-production operator, even in small sequences, the opportunity to easily modify viewing trajectories, introduce computer generated elements, stabilise jumpy action or positionally justify action in the view port, create alternative camera trajectories and add new static details into the background sequence [Giaccone, Greenhill and Jones, 1998]. For instance, custom advertisements can be easily added to the background of a recorded scene, or the same person can be composited into the final video sequence more than once [Bender, 1993].

A representative application that exploits the unlimited resolution of mosaics is in digital recording of paintings, where it is required to store a picture in its highest detail but the format of the image sensor is not large enough to provide the necessary resolution [Gümüstekin and Hall, 1996b]. Particularly useful can be the construction of mosaics using off-the-shelf inexpensive PC cameras. High quality stills and omnidirectional

views can be created without requiring any special (and costly) equipment [Sawhney and Kumar, 1997].

There are a number of commercial products using image-based rendering systems. Some ([QuickTime VR], [Surround Video] and [Spin Panorama]), use cylindrical images with a limited vertical field of view (e.g. images 2496 by 768 pixels in size are used in QuickTime VR applications).

Certain panorama programs (e.g. Quick Time VR) can assemble a panorama from a series of still photos taken with an ordinary camera. The alignment between overlapping images is not fully automatic, and does require some intervention by the developer. Among other things, the program must be instructed whether the camera was oriented vertically or horizontally (to include as much vertical space as possible the camera is usually turned sideways) and how much overlapping area between frames should be analysed when the edges are joined together. In other programs the image alignment is fully manual (e.g. Spin Panorama). In all cases the programs blend exposure differences between overlapping frames. Surround Video, on the other hand, requires a special panoramic camera that records a panoramic image in one single piece of film (these will be discussed in 2.3.1), eliminating the mosaicing process.

Panorama viewers are used to render distortion-free views of portions of the panoramas, according to user input. Any portion of the 360° panoramas can be viewed: the image on the screen represents a small arc of the entire 360° (usually about 70°). A virtual camera pans the image left or right in a complete circle and zoom in for a closer look at small details, although vertical scrolling is limited to the field of view captured by the camera's lens. Panoramas are interactive in ways that are impossible without a computer. For instance, developers can link panoramas together, so users can change viewpoints by clicking on predefined hot spots or examine certain objects in more detail.

Newer systems support full spherical maps (e.g. [Omnecam], [Smooth Move], [Be Here], and [RealVR]), although these programs require omnidirectional cameras (see 2.3.2) to acquire the images or, alternatively, computer generated images. Panoramic videos are an exciting experience that has become possible using a personal computer. In a panoramic

video, each image of the video sequence is an omnidirectional view. The user can move in a predetermined path through the scene while looking in any direction.

1.2 Aim of the Research

This thesis describes research into the problem of mosaic construction. Composing mosaics from a number of overlapping images is a problem of image registration. Existing registration algorithms are designed to align pairs of images, but cannot be used with more than two images simultaneously. It will be shown that the accumulation of small alignment errors may become apparent in the mosaic regardless of the accuracy attained in the registration process.

The prime motivation of the research project described in this thesis was to develop new techniques for the construction of mosaics. Since error-free image registration techniques do not exist, it is the intention of the author to develop a new technique, based on existing imperfect registration methods, for consistently aligning all images which form part of a mosaic.

In achieving the solution, a new type of mosaic will be possible. The spherical mosaic, composed from images obtained by panning and tilting a camera, represents an omnidirectional view of the scene.

1.3 Overview of the Thesis

The thesis is organised to present material in a progressive manner. It begins by introducing the concept of a mosaic and its applications.

Chapter 2 examines the operation and limitations of the conventional camera when used for the acquisition of images for mosaic construction. Subsequent sections discuss a set of specialised panoramic and omnidirectional cameras.

In Chapter 3, image registration and image integration techniques are described as the basic steps involved in mosaic construction. The types of mosaics that can be created for each case of camera motions are also presented.

Chapter 4 introduces the *Looping Path Problem*, and a discussion of previously proposed solutions. A novel solution is offered which uses a physical analogy to distribute the accumulated alignment error in the mosaic. The concepts and equations for a physical simulation are reviewed and a physical model for the mosaic is presented.

Chapter 5 describes the implementation of the image registration technique adopted in this project. Results on planar scene mosaics are presented.

Chapter 6 discusses, in three progressively complex experiments, the constraints developed for the physical model of the spherical mosaic. In the last experiment the results on the spherical mosaic are presented.

In Chapter 7, conclusions for this thesis are presented. This summarises the original contributions to knowledge made in this research, and recommends improvements and future directions which can be investigated.

In this thesis the following convention is used: variables printed in bold denote vectorial magnitudes (e.g. \mathbf{a} , \mathbf{F}), where variables in italics (e.g. m) are scalar quantities. Bold and italics are used for matrices (e.g. \mathbf{S}_{body}).

Chapter 2 IMAGE ACQUISITION

2.1 Introduction

In conventional photography, much thought must be given to choice of viewing position, composition, and lighting, while at the scene. This remains true for panoramic image acquisition, with the additional complexity of a wider and generally more assorted scene to be recorded. This chapter explains the operation of the conventional camera and its limitations. Also a set of specialised cameras designed specifically to record panoramic and omnidirectional views is discussed.

2.2 Conventional Camera Technology

Although imaging technology has advanced enormously in the past few years, there are still limitations in image acquisition which are necessary to understand. Obtaining images from the real world requires setting a number of variables to accommodate to each particular scene. Light conditions, distance of objects to the camera, and the motion of both, affect the camera settings if a sharp, focused image is to be obtained.

The following sections describe the variables involved in the image acquisition process and their interrelation. These apply to both still and video cameras, which differ only in sensor type and the fact that video cameras record images repetitively.

2.2.1 Image Sensor

In a still camera, the imaging device is the film. In video and digital cameras, the imaging sensor is generally a *Charge-Coupled Device* (CCD). A CCD is a two-dimensional array of transistors sensitive to light, which convert the light intensity into electrical signals. Each of these transistors corresponds to a pixel in the image. Colour sensors are implemented either using three CCDs, one for each colour component (red,

green and blue), or using one CCD where, for each pixel, the transistors are arranged into groups of three, each transistor sensitive to a colour component.

2.2.2 Field of View and Focal Length

The *field of view* (α) is the coverage angle of a camera (see Figure 2-1). In a conventional camera, the horizontal field of view is wider than the vertical field of view. The *focal length* (f) is the distance between the sensor plane and the optical centre of the lens. The zoom or magnification of the lens is modified by varying the focal length. The field of view and the focal length are related by the following expression,

$$\alpha = 2 \tan^{-1} \left(\frac{h}{2f} \right) \quad (1)$$

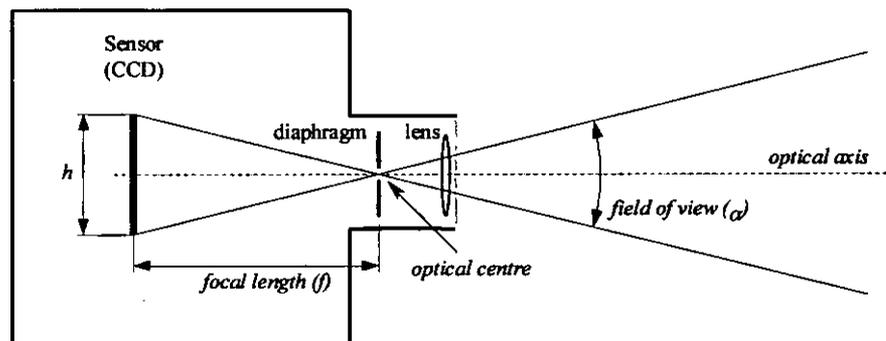


Figure 2-1 Simplified camera model showing the field of view and the focal length.

2.2.3 Lens Aperture

The *diaphragm*, also called *iris*, is a device for controlling the amount of light passing through the lens system. This is constructed from a series of flat, interleaving metal blades, which together form a central hole of adjustable size. The area of this *aperture* controls the brightness of the image the sensor receives from the scene. It can be changed either manually by turning a ring on the lens barrel or electronically in cameras provided with this feature.

The influence of the lens aperture is twofold:

It determines how much light from the scene reaches the sensor.

It affects the depth of the scene that appears sharp and will be explained in section 2.2.4.

In general, the lens aperture is determined principally by the brightness of the scene. Strong sunlight may require a reduced aperture, dim light demands a wider aperture.

2.2.4 Depth of Field

The camera lens can focus sharply only for one distance at a time. Nearer and farther than that focused plane, the definition (clarity) deteriorates until eventually detail can be no longer discerned at all. However, objects nearer and further than the focused distance still appear quite well-defined. This region of sharp reproduction is known as the *depth of field*. At larger lens apertures, the lens gives less depth of field, at smaller apertures, this sharp zone steadily increases.

The scene before the camera is usually three-dimensional; it has depth. So, as the subject (and/or the camera) moves within the scene, the camera's focus control needs to be adjusted continually to keep the selected part of the scene sharp.

Fixed-focus cameras represent a simplified compromise design. A fixed-focus camera has its focus set to the hyperfocal distance. Everything will be reasonably sharp from half this distance (nearest point), away to infinity. The *hyperfocal distance* is proportional to the square of the focal length and to the lens aperture, i.e. a short hyperfocal distance is achieved with a small focal length (i.e. large field of view) and a small lens aperture (see Figure 2-2). As a fixed-focus camera only works with a comparatively small aperture to provide depth of field, light intensities must be high. Besides, for short-distance subjects (i.e. close shots), the fixed-focus camera becomes unacceptably defocused. However, they make operation easier, and provide an overall clarity that is sufficient for many applications.

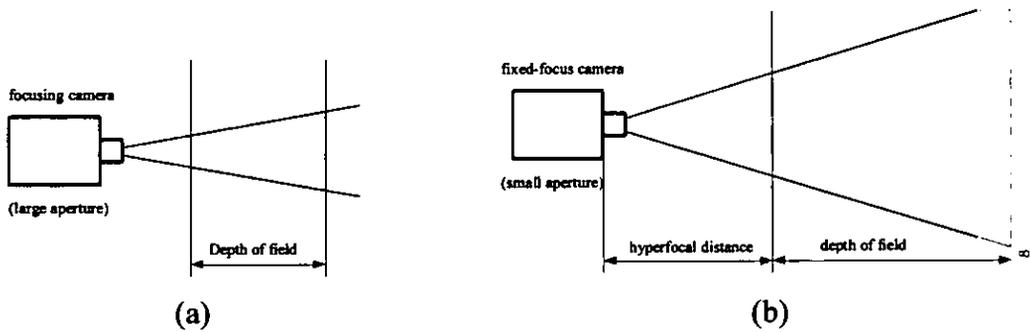


Figure 2-2 Objects within the depth of field will appear in focus. (a) Focusing camera. (b) Fixed-focus camera.

2.2.5 Shutter Speed

Shutter speed, also called *exposure time*, refers to the length of time the sensor is exposed to the light from the scene for one frame. For film, the shutter is a mechanical device, while in a CCD the shutter speed is controlled electronically, common speeds ranging from 1/50 s up to 1/2000 s.

Large exposure times will cause moving objects, or the entire scene in the case of a moving camera, to appear blurred (*motion blur*). When photographing fast moving scenes, a small exposure time is a must. The exposure time is also determined by the brightness of the scene. Bright scenes may require less exposure time than dim scenes. The exposure time is normally set in conjunction with the lens aperture to accommodate to a particular scene brightness. Any change in the lens aperture must simultaneously affect the exposure.

2.2.6 Video Fields

An interlaced video frame is composed of two fields, each one containing half of the scan lines needed to make up one video frame. That is, the first field contains every other scan line of the frame, and the second field contains the rest of the lines. A large TV screen that is updated at 25 frames per second would flicker, i.e. the image would begin to fade away before the next one was drawn on the screen. By using two fields, each containing one half of the information that makes up the frame and the fields being drawn on the screen consecutively, the field update is 50 fields per second. At this update rate, the eye blends everything together into a smooth, continuous motion.

In the acquisition process, the video camera registers the fields sequentially (i.e. one field after the other), and since there is a lag between the field acquisitions, motion artifacts may occur when the two fields are displayed together. A common procedure to achieve a motion-free image is to use only one field per frame and discard the other, although this reduces the available resolution by half.

Fields only exist for interlaced scanning systems, that is, most video cameras except progressive scan video cameras, which are designed to capture the whole frame at once, and digital still cameras, which register non-interlaced images.

2.2.7 Automatic Gain Control (AGC)

Video cameras and digital cameras can adjust the amplification of the signal received from the sensor (CCD) to accommodate different illumination conditions. The intensity is integrated over the image and this information is used to automatically adjust its brightness. If dark and bright scene elements are combined in the same image (i.e. large dynamic range), for instance, an image of a dim room with a blazing vista of snow peaks and clear blue sky seen through a window, the contrast and therefore the details of the image are reduced, particularly in the very bright and dark regions. For the same scene, another view of the room without the window will present distinctive detail.

2.2.8 Lens Distortion

The lens is a complex item that has inherent distortions and its quality is directly related to cost. Lens aberrations are the deficiencies in image due to the lens, and include spherical aberration, coma, astigmatism, curvature of field, distortion of the image and chromatic aberration [White, 1982]. Distortion of the image (barrel/pincushion) is due to the magnification of the lens being dependent upon the direction of the incident rays. *Coma* is caused by the differences in the power of various annular zones of the lens (i.e. from the thick to the thin part) and occurs when the parallel rays are at an angle to the principal axis. This results in the image produced by the marginal rays having a reduced brightness.

2.2.9 The Ideal Camera

The camera model generally used in computer vision is the ideal pinhole camera [Mann and Picard, 1994c], [Bender, 1993], [Gümüstekin and Hall, 1996b]. It assumes infinite depth of field, and that light travels in straight lines (ignores diffraction).

The use of this model can be justified when first, the objects in different depths recorded by the camera are in focus (which is the case when the diameter of the aperture of the camera is very small). Second, there is no significant distortion caused by physical characteristics of the optical elements (i.e. no barrel/pincushion distortion). And third, the distance from the camera to the objects is much larger than the distance between the film plane and the diaphragm.

The camera projects a 3D scene onto a 2D plane (see Figure 2-3). If the coordinates of the objects in the scene is known, say (x_0, y_0, z_0) , the corresponding point in the image (x_i, y_i) is given by:

$$x_i = \frac{-f x_0}{z_0 - f}, \quad y_i = \frac{-f y_0}{z_0 - f} \quad (2)$$

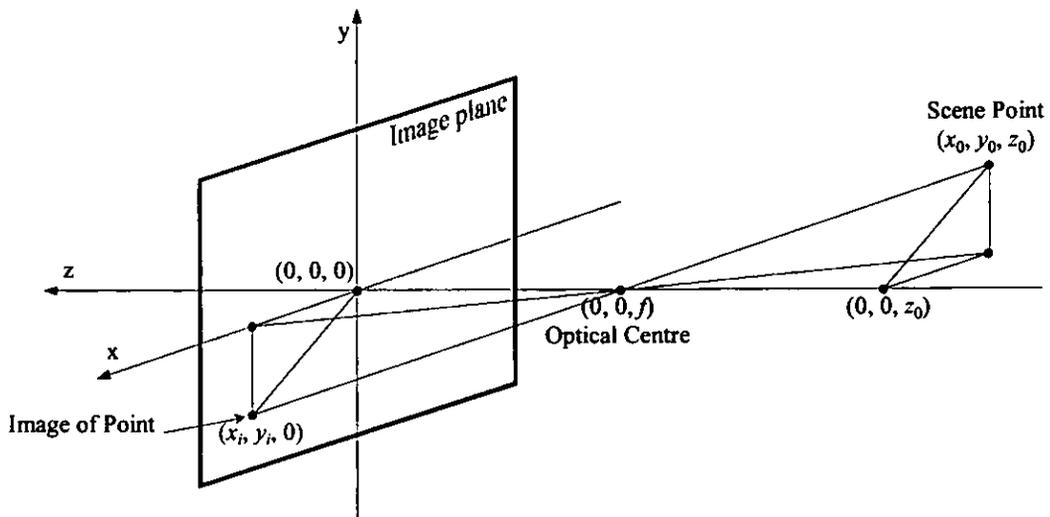


Figure 2-3 Pinhole camera model.

2.2.10 The Human Eye

The human visual system is one of the most complex in existence. It consists of an eye that transforms light to neural signals (see [Lim, 1990] for a more comprehensive description). Functionally, the eye is a device that gathers light and focuses it on its rear surface. From the point of view of image mapping, the most important difference between the human eye and a camera is the shape of the surface of projection. In the eye, it is approximately spherical, while a camera sensor is planar. The projection of images onto a planar sensor yields to projective distortion (see Figure 3-1d), which is not applicable to the spherical projection surface of the human eye.

2.3 Special Devices for Panoramic Image Acquisition

There are several variations of the conventional camera worth mentioning in this thesis (panoramic cameras, omnidirectional cameras and non-frontal imaging cameras). Either enlarging the field of view, or reducing the focusing limitations of a conventional camera, they improve the acquisition process of immersive images. A discussion on the suitability for different applications follows the description of these devices.

2.3.1 Panoramic Cameras

Panoramic cameras are specially designed to record a cylindrical panoramic image onto a long film strip [Meehan, 1997]. A panoramic camera automatically rotates about a fixed nodal point (i.e. pans 360°) while continuously exposing a strip of film that moves in the opposite direction.

The vertical field of view is not very wide in panoramic cameras. This may be too restrictive for some applications, and only the most expensive models have interchangeable lenses. To digitise the long film strip from a panoramic camera with enough resolution to support high magnifications, an expensive drum scanner that is big enough to accommodate the strip must be used [Halfhill, 1995]. Standard film scanners cannot be used, and flatbed scanners are not recommended.

Although the use of a panoramic camera simplifies the acquisition of panoramic photographs, in the digital era alternatives have been proposed to reduce cost and limitations.

2.3.2 Omnidirectional Cameras

An omnidirectional camera can record an omnidirectional view (full 360° vertical and horizontal field of view) from a single viewpoint. Its implementation uses a conventional camera and a parabolic or pyramidal mirror, which maps a 180° hemispherical view onto one single image. For a 360° full view, two cameras facing in opposite directions are used.

[Nayar, 1998] developed a hemispherical video system, called *omnicam*, combining a standard lens on a video camera with a paraboloid mirror. Two such cameras can be placed back-to-back, without violating the single viewpoint constraint, to achieve an omnidirectional sensor. The only blind spot is at the apex of the parabola where the camera sees itself. Another very wide field of view lens system has been introduced by Be Here Corporation [Be Here], named *Portal* lens. It captures a 360° by 100° aspect ratio image with a single camera [Hamit, 1997]. It uses one large parabolic mirror, and four optical elements (i.e. lenses), which then feed the image to a standard Nikon 24mm camera lens. Images recorded by an omnidirectional camera must be unwrapped in software or projected onto a hemispherical dome, to produce a distortion free view of the scene.

In the creation of immersive images, omnidirectional cameras surpass mosaicing techniques in several ways: firstly, by eliminating the composition process of images and possible misalignments and visible seams. They also avoid problems associated with parallax, motion blur and moving objects, and, most important, they can operate in real-time, which makes omnidirectional films possible, with each frame containing a full view.

However, due to the mapping of a hemispherical view onto a flat image the resolution of an omnidirectional image is four times as great for objects seen near the centre of the mirror than at the edges. This inefficient distribution requires a sensor with four times the

useful final resolution. The developers of the "Be Here's Portal" lens recommend using film as an intermediate medium. Film has a higher resolution than most digital cameras (there are some high end cameras that offer better resolution, but they are very expensive), and as much resolution as possible is needed [Hamit, 1997]. On the other hand, mosaics offer essentially unlimited resolution. One of their most attractive applications is the creation of omnidirectional views using off-the-shelf inexpensive PC cameras.

Moreover, mosaicing possesses a range of applications of its own, for which omnidirectional cameras are not suitable. These include video compression and the complete and efficient representation of large video streams [Irani, Anandan and Hsu, 1995]. In the video and film post-production industry, it offers the operator the opportunity to easily modify viewing trajectories, introduce computer-generated elements, stabilise jumpy action, positionally justify action in the view port, and add new static details into the background sequence [Giaccone, Greenhill and Jones, 1998].

2.3.3 Non-frontal Imaging Camera

A non-frontal imaging camera (NICAM) is a camera in which the sensor plane is tilted. Since the sensor plane is at a non-perpendicular angle to the optic axis, different parts of the sensor are focused at different focus depths [Krishnan and Ahuja, 1996].

To create fully focused panoramic images requires individually focused images at every pan angle. For scenes with bright light conditions, the lens aperture can be made small enough to increase the depth of field to near infinity, and using a frontal camera (i.e. conventional camera) is the easiest way to create a panoramic mosaic.

However, if infinite depth of field is not possible (i.e. dark scene or close objects), parts of the scene will appear defocused. Then panning a NICAM is an alternative to obtain a focused panoramic image. The same object will appear in different images and in different focus. For the composition of the mosaic, only the focused areas of the images are used. [Krishnan and Ahuja, 1996] explains a method for the composition of panoramic mosaics using a non-frontal camera.

For scenes where a conventional camera needs to be focused for each image composing the panorama, then using a non-frontal camera might reduce the acquisition time, although it introduces additional post-processing complexity. For a conventional fixed-focus camera, only dark scenes, and objects too close to the camera are problematic. Practically, most scenes are correctly acquired in focus, in the worst case, at the expense of increasing the exposure time.

2.4 Summary

Understanding the operation and limitations of the real camera is important in image processing applications. Despite technological advances in image acquisition devices, the art of photography, whether using film, digital format or video, requires the photographer to decide on a number of camera parameters (field of view, lens aperture, shutter speed...), to accommodate for different scene conditions.

This chapter has covered the interrelation between the factors determining the quality of images in the acquisition process with real cameras. A set of specialised cameras designed specifically to record panoramic and omnidirectional views was discussed and their limitations exposed. In next chapter, techniques for mosaic composition will be presented.

Chapter 3 MOSAIC CONSTRUCTION

3.1 Introduction

Mosaicing is the technique used to produce large images by assembling two or more individual overlapping images. The resulting image is called a mosaic, and it has a wider field of view and higher resolution than each of the individual images.

The basic processing steps involved in mosaic construction are well known [Peleg and Herman, 1997a], and can be summarised as follows:

Image registration: Determines the transformation that aligns every pair of overlapping images. The quality of a mosaic directly depends on the quality of the individual alignments between overlapping images. The displacements must be estimated to high precision so that the images can be positioned in the mosaic to a small fraction of a pixel.

Image integration: Consists of the selection of non-overlapping areas in the images that will contribute to the final mosaic or the combination of pixel intensities from overlapping images. Further blending of neighbour images is necessary to reduce the visibility of seams due to differences in intensities.

In this chapter, the steps and techniques involved in the composition of mosaics are described.

3.2 Image Registration

Registration is a fundamental task in image processing used to match two or more images taken, for example, at different times, from different sensors, or from different viewpoints [Brown, 1992]. A transformation must be found so that the points in one image can be

related to their corresponding points in the other image. For two images, denoted by I_A and I_B , where $I_A(x, y)$ and $I_B(x, y)$ each map to their respective intensity values, the mapping between the images can be expressed as

$$I_A(x, y) = I_B(f(x, y)) = I_B(x', y') \quad (3)$$

where f is a 2D spatial-coordinate transformation, that is, a transformation which maps two spatial coordinates, x and y to new spatial coordinates x' and y' . Thus, image registration is the process of finding the transformation (i.e. function f) that best aligns two images.

Over the years, a broad range of image registration techniques has been developed for various types of problems, such techniques are reviewed in [Brown, 1992]. Particularly, registration techniques have been applied to image mosaicing by [Mann and Picard, 1994c], [Irani, Anandan and Hsu, 1995], [Chen, 1995] [McMillan and Bishop, 1995] and [Szeliski, 1994b] among many others.

3.2.1 Transformation Models

In the simplest case, two images can be related by a pure translation. This is followed in complexity by translations and rotations (rigid transforms, Figure 3-1b), then scaling, i.e. similarity transforms, affine transforms (see Figure 3-1c), and full projective transforms (Figure 3-1d) (see [Wolberg, 1990] for a detailed survey on transformations). Several authors have suggested using projective transformations for registration of the images in a mosaic [Mann and Picard, 1994c] [Irani, Anandan and Hsu, 1995] [Szeliski, 1994b] [Gümüstekin and Hall, 1996b].

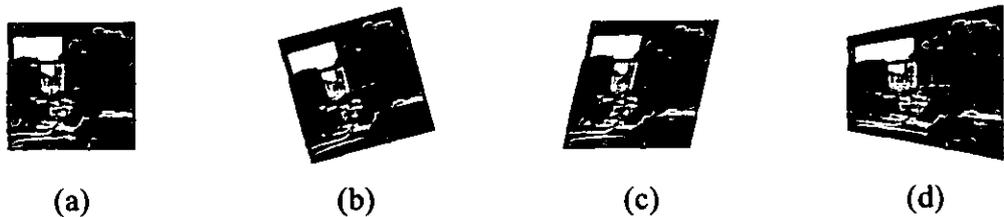


Figure 3-1 (a) Original image. (b) Rigid, (c) Affine, and (d) Projective transformations.

The mapping or transformation $f()$ can be expressed in matrix form using homogeneous coordinates to represent points, that is, 2D points in the image plane are denoted as (x, y, w) [Szeliski, 1994a]. The corresponding Cartesian coordinates are $(x/w, y/w)$. The 2D planar transformations can be described as follows:

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix} \quad \text{or} \quad f(x, y, w) = \mathbf{M} (x \ y \ w)^t \quad (4)$$

where x, y are pixel coordinates,

$x'/w', y'/w'$ are the new pixel coordinates,

m_{ij} are the parameters that describe the transformation.

For each case (i.e. rigid, similarity, affine and projective transformation), a different number of parameters m_{ij} are necessary to describe the transformation [Szeliski, 1994a].

A *rigid transformation* model involves only three parameters: two for the translation ($\Delta x, \Delta y$) and one for the rotation ($\Delta\theta$). The rigid transformation matrix is as follows.

$$\mathbf{M}_{\text{rigid}} = \begin{pmatrix} \cos \Delta\theta & -\sin \Delta\theta & \Delta x \\ \sin \Delta\theta & \cos \Delta\theta & \Delta y \\ 0 & 0 & 1 \end{pmatrix} \quad (5)$$

where $\Delta x, \Delta y$ are the translation components and $\Delta\theta$ is the angle of rotation (as shown in Figure 3-2).

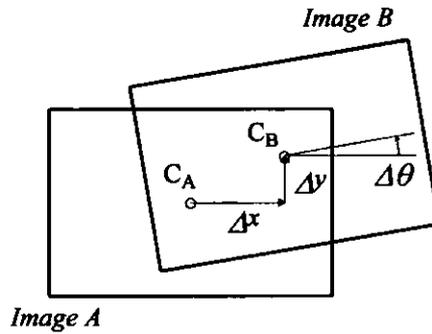


Figure 3-2 Rigid transformation parameters that align two overlapping images. C_A and C_B are the centres of images A and B respectively.

The *affine transformation* model involves six parameters (6), while the *projective transformation* model has a general matrix \mathbf{M} (4) with 8 degrees of freedom. Note that two matrices are equivalent if they are scalar multiples of each other. This redundancy can be removed by setting $m_{33}=1$ [Szeliski, 1994a].

$$\mathbf{M}_{\text{affine}} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{pmatrix} \quad (6)$$

3.2.2 Camera Motion

Mosaics can be created from video sequences recorded by a variety of camera motions. There is a connection between the constraints imposed on the camera motion and the number of parameters involved in the transformation matrix. Hence, for each type of camera motion, a suitable transformation model is required.

The determination of the optimal transformation model for registration depends on the types of variations between the images. [Mann and Picard, 1994c] identify the cases where the *affine model* and the *projective model* correctly describe the relationship between two images from a video sequence.

The *affine model* applies to:

Case 1. A camera free to zoom and to rotate about its optical axis on an arbitrary static scene.

Case 2. An arbitrary planar scene –such as a document or a whiteboard– where the camera is free to zoom, to rotate about its optical axis and to translate, provided that the plane of the image sensor remains parallel to the planar scene (see Figure 3-3). The affine model, however does not correctly account for camera pan or tilt.

The *projective model* correctly describes two broader cases:

Case 3. Planar scene free to move arbitrarily, and camera free to move arbitrarily (see Figure 3-4).

Case 4. A camera at a fixed location, free to zoom and to rotate about its optical centre of projection (i.e. camera free to rotate about its optical axis, and to pan and tilt), on an arbitrary static scene (see Figure 3-6). The particular case where the camera can pan but it is not allowed to tilt is represented in Figure 3-5.

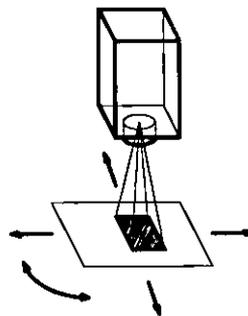


Figure 3-3 Camera fixed over a desk with optical axis perpendicular to desk surface. As a document is translated and rotated under the camera, different portions of it become visible. Any two frames are related by a translation and a rotation and scale if allowed to zoom.

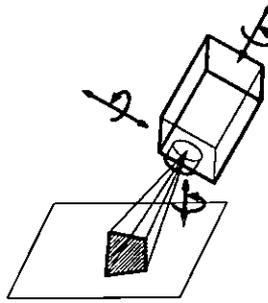


Figure 3-4 Planar scene and camera free to move arbitrarily. For a camera allowed to change its position (i.e. translation), its orientation (i.e. rotation) and zoom over a planar scene, two overlapping frames are related by a projective transformation.

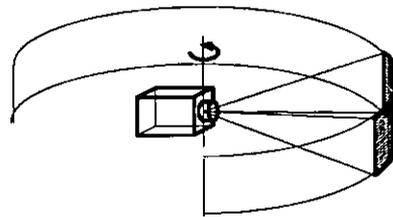


Figure 3-5 For a camera free to pan about its optical axis, the resulting mosaic can be pasted onto a cylinder.

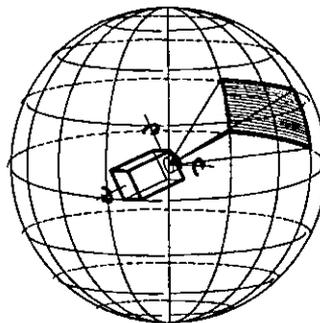


Figure 3-6 Camera free to zoom, pan and tilt about its optical centre of projection. The resulting mosaic can be pasted onto the interior surface of a sphere.

3.2.3 Image Registration Techniques

[Brown, 1992] classifies image registration techniques according to four major components: feature space, similarity measure, search space and search strategy. Every registration technique can be thought of as a selection for each of these components.

3.2.3.1 Feature Space

The feature space extracts the information in the images that will be used for matching. They can be the pixel intensities, but other features are commonly used such as edges, contours, surfaces, salient features such as corners [Zoghiani, Faugeras and Deriche, 1997] [Capel and Zisserman, 1998], text [Zappala, Gee and Taylor, 1997], lines [Janssen and Vossepoel, 1994], and points of high curvature. Also statistical features such as moment invariants or centroids, and higher-level structural and syntactic descriptions can be used.

The choice of feature space depends on the content of the images to align, since the features to be identified and matched should, first of all, be embodied in the images. When no information on image content is known in advance, such as in a general mosaic application, the use of any particular salient feature and the risk of its absence should be avoided. On the other hand, *direct estimation*, i.e. based on pixel intensities, has proven to be more practical and robust over feature correspondence based methods since the direct methods typically use all the data and structure constraints in the estimation process [Sawhney and Kumar, 1997].

3.2.3.2 Similarity Measure

The similarity measure determines the relative merit for each test. Typical similarity measures are cross-correlation, sum of absolute differences, and Fourier invariance properties such as phase correlation. Using curves and surfaces as a feature space requires measures such as sum of squares of differences between nearest points.

Cross-correlation measure is the basic statistical approach to registration. By itself, cross-correlation is not a registration method. It is a similarity measure of match metric, i.e. it gives a measure of the degree of similarity between two images. However, there are several registration methods for which it is the primary tool. The cross-correlation

measure is directly related to the more intuitive measure which computes the sum of the differences squared between the two images at each pixel location:

$$D(f) = \sum_y \sum_x [I_A(x, y) - I_B(f(x, y))]^2 \quad (7)$$

This measure decreases with the degree of similarity since, when the images are superimposed using the transformation $f()$ for which the two images are most similar, the differences between the corresponding intensities will be smallest. A more efficient class of algorithm was proposed by Barnea and Silverman [Barnea and Silverman, 1972]. They suggested a similarity measure (8) which is computationally much simpler, based on the absolute differences between the pixels in the two images:

$$D(f) = \sum_y \sum_x |I_A(x, y) - I_B(f(x, y))| \quad (8)$$

Phase correlation has been used to find large displacements between overlapping images [Szeliski, 1994b]. This technique estimates the 2D translation between a pair of images by taking 2D Fourier transforms of each image, computing the phase difference at each frequency, performing an inverse Fourier transform, and searching for a peak in the magnitude image.

3.2.3.3 Search Space

The search space is the set of transformations that is capable of aligning the images. It is the space of all possible transformations. Each transformation candidate is evaluated using the similarity measure. The search continues according to a search strategy until a transformation is found whose similarity measure is satisfactory.

3.2.3.4 Search Strategy

The search strategy decides how to choose the next transformation from this space, to be tested in the search for best transformation. Because of the large computational cost associated with many of the matching features and similarity measures, all possible

transformations can not be tested, and a good search strategy reduces the computational cost of finding the best match.

To reduce the search space, a search strategy such as gradient descent or Levenberg-Marquardt can be used. Unfortunately, they both only find locally optimal solutions. If the motion between successive frames is large, a different strategy must be used to find the best registration [Szeliski, 1994a].

Another technique commonly used in image processing (image registration, pattern recognition, classification, etc.) to achieve shorter computation times is *hierarchical matching* (also called pyramidal structures and Laplacian pyramids) [Bergen et al., 1992] [Hansen et al., 1994]. Smaller images are created from the original images by averaging blocks of pixels as shown in Figure 3-7. The use of pyramidal data structures in image registration permits the alignment at coarse resolutions before proceeding with the full resolution original images. The best transformation is found for the small images. This transformation is then used as an initial position to compute the transformation of the full scale original images.

The computation of the similarity measure is an operation the expense of which increases as the square of the resolution at which it is performed, therefore, reducing the resolution of the images improves significantly the computation time. Furthermore, the apparent motion is also smaller in these small images, therefore the search space is also reduced.

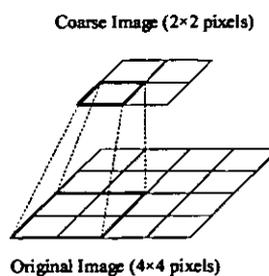


Figure 3-7 Image as a pyramidal data structure. A succession of levels vary the resolution at which the data is represented.

Image processing hardware and algorithms to provide real time alignment have been recently achieved. David Sarnoff Research Center has developed an image processing

system based on multiresolution, pyramid, pipeline architecture tailored to the image alignment task at video rates [Burt, Hansen and Anandan, 1996]. The system can compensate for translation, rotation, dilatation and skew (i.e. affine) components of image motion. Alignment is typically accurate to 1/10 pixel. In the search strategy, it combines the pyramidal approach with the use of progressively complex transformation models, that is, finding first only translations at small resolution pyramid levels, and using this as an initial estimate for the affine model.

3.2.4 Unwanted Elements

Several factors have been identified that deteriorate the quality of image registration. These include motion parallax, lens distortion, moving objects and motion blur.

3.2.4.1 Motion Parallax

For a camera translating in its sensor plane, objects close to the camera translate more than objects farther from the camera. Occlusions occur. As a result, a transformation that will align a close object will duplicate far objects, and on the other hand, a transformation that will align a far object will truncate closer objects. In addition, trying to align close and far objects with a single transformation might yield useless results.

When parallax is involved no single transformation can be found to represent the optical flow in the entire scene. However, parallax problems can be overcome by synthetically generating intermediate images [Rousso, Peleg and Finci, 1997b]. New views can be synthesised using various methods, such as optical flow interpolation [Chen and Williams, 1993] [Seitz and Dyer, 1995]. In [Seitz and Dyer, 1995], a simple image rectification procedure is described which guarantees that interpolation does in fact produce valid views, under generic assumptions about visibility and the projection process. Issues on motion parallax as well as forward motion and zoom are addressed in [Rousso, Peleg and Finci, 1997a].

3.2.4.2 Lens Distortion

The distortion introduced in the acquisition process by the camera lens (see section 2.2.8) should also be modelled and corrected if accurate alignments are to be obtained. The difference will usually manifest itself primarily as relative geometric distortion (i.e.

barrel/pincushion) between the two images that makes it impossible to exactly overlay the images everywhere simultaneously. As a result the height of the cross-correlation peak is reduced and image registration performance degrades.

The effects of geometric distortion are shown to depend on the size and shape of the images relative to the size and shape of the cross-correlation function [Mostafavi and Smith, 1978]. The more severe is the geometric distortion, the smaller is the optimum size. Where lens distortion is not modelled, in order to minimise its detrimental effects, the matching should be computed using image areas closest to the centres of the images.

One application of mosaic techniques is the creation of omnidirectional views using off-the-shelf inexpensive PC cameras. Severe lens distortion is a common occurrence in most of these cameras. In order to create high quality mosaics using these cameras, it is necessary to correct the distortion. [Sawhney and Kumar, 1997] have proposed a method for image alignment which automatically corrects for lens distortion.

3.2.4.3 Moving Objects

When image registration is performed assuming a static scene, unwanted moving objects—such as the effect of the wind on trees and clouds—can cause misregistration between images, and a registration technique robust enough to withstand the influence of moving objects must be used.

When the motion of particular objects is to be preserved as an essential characteristic of a video sequence, these can be identified and isolated using image processing. It is fairly common in video sequences that a mostly fixed background is present with or without independently moving objects. The dominant background in the image plane changes mostly due to camera operations and motion (zoom, pan, tilt, track etc.) [Sawhney, Ayer and Gorkani, 1995]. Moving objects can then be eliminated from the mosaic, or they may be inserted at regular time intervals in order to suggest their trajectory over the scene [Irani, Anandan and Hsu, 1995].

3.2.4.4 Motion Blur

Motion blur is caused by the variations of the image projected on the camera sensor during light integration (i.e. the time the sensor is exposed to the light). Large inter-frame motions produce motion blur. Hence with fast shutter speeds (i.e. short exposure time), the effect of motion blur is reduced (see section 2.2.5). Although the faster the shutter speed, the darker the image, and either the iris must be opened or the illumination has to be increased to compensate.

3.3 Image Integration

Once the transformations that align pairs of images have been computed, the next step is to integrate the images in the mosaic.

The alignments are usually better near the centre of the images than at the corners, particularly if lens distortion has not been corrected, since the geometric distortions from the lens (i.e. pincushion/barrel) will be less pronounced near the centre of the images and more exaggerated at the corners. A simple procedure to render the mosaic is to copy, from each image, only the pixels which are closer to its centre than to any other image centre. This scheme corresponds to the *Voronoi tessellation* (discussed in more detail in section 3.4), which, given the position of the centres of the images, defines a polygonal area for each image to be pasted on the mosaic. Another way of deciding the contribution of each image to the mosaic is to find a *best frontier* that minimises visibility of the seam along the overlapping regions of the images [Milgram, 1977].

3.3.1 Image Blending

Unfortunately, intensity differences between the images due to camera automatic gain control (see section 2.2.7) or lens aberrations such as coma (darkening of the corners, see section 2.2.8) may introduce visible artefacts. The visual effect may be very noticeable, and it often interferes with the perception of the details of the picture.

Several methods have been developed to eliminate the seams while preserving the details and the sharpness. The case where two images have an overlapping area was studied by [Milgram, 1975] [Milgram, 1977]. Milgram suggested that *shifting the histogram* can

help to match the intensity levels in the common area of the two images. However, intensity level shift alone does not always account for the visible intensity differences between the images. Further blending of neighbour images is therefore necessary to reduce the visibility of seams due to differences in intensities and possible small misalignment. In advertisement, image blending has been numerously applied to create fancy synthetic images from possibly unrelated components.

Other techniques are based on smoothing both sides of the seam by a relaxation algorithm [Peleg, 1981] i.e. an average or a median of the intensity values of the aligned images. *Weighted average* or *weighted median*, where the weights can be chosen to decrease with the distance of a pixel from its image centre, has the additional advantage of accounting for alignment inaccuracies near image boundaries [Irani, Anandan and Hsu, 1995]. With appropriate choice of weighting function within a transition zone, the weighted average technique will result in smooth transition, but an invisible seam is not ensured [Hsu and Wu, 1996].

Multiresolution technique [Burt and Adelson, 1983a] has been proposed for image blending, in which Laplacian pyramid [Burt and Adelson, 1983b] is used, and a weighted average within a transition zone. The images are decomposed into several frequency bands. The weighted average is then performed for each band choosing the width of transition zone proportional to the wavelength represented in the band. That is, for lower frequency components, the width of transition zone is chosen to be larger than that of higher frequency components. This implies that low-frequency components extend across the boundaries further than high-frequency components. In psychophysics and the physiology of human vision, evidence has been gathered showing that the retinal image is decomposed into several spatially oriented frequency channels. The weighted average concept using different transition zones according to different frequency bands quite fits human visual sensibility [Hsu and Wu, 1996].

The multiresolution pyramid provides a particularly effective framework for image merging, eliminating seams at all scales: coarse image patterns are blended over a long distance while fine patterns are blended over a proportionally shorter distance.

Other integration schemes provide image enhancement, such as super-resolution [Irani and Peleg, 1991], which reconstructs a high-resolution image from a number of lower resolution, possibly noisy, images of the same scene where the successive images are uniformly shifted versions of each other at subpixel displacements [Tekalp et al., 1992].

3.4 Voronoi Diagrams and Delaunay Triangulation

The Voronoi diagram is named after the mathematician M. G. Voronoi who explored this geometric construction in 1908 [Voronoi, 1908]. However, as early as 1850 another mathematician, G. L. Dirichlet, studied the problem [Dirichlet, 1850]. Accordingly the Voronoi diagram is sometimes named Dirichlet tessellation.

The Voronoi diagram is one of the most fundamental and useful constructs defined by irregular lattices. It has proved to be particularly useful in modelling natural phenomena [Aurenhammer, 1991]. As it is a widely used geometrical construction, the diagram has several names. The construction is utilised in many distinct fields, and is often named after the person who first used it in a particular field. In geography they are known as Thiessen polygons, from the climatologist A. H. Thiessen; in metallurgy, Wigner-Seitz cells; and in biological shape and visual science, Blum's transform. As a result, the vocabulary in this area is redundant and disjoint.

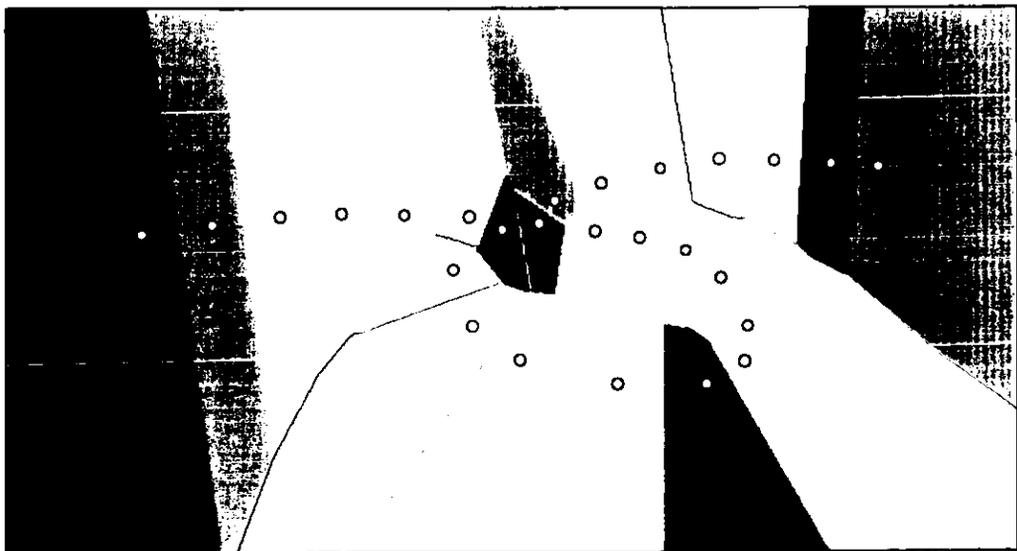


Figure 3-8 Voronoi diagram. The points are represented in white, and the polygons in different colours.

The *Voronoi diagram* of a set of points P is a set of polygonal areas Ω . For each point p_i , a polygonal area $\omega(p_i)$ is defined such that every point contained in this polygonal area is closer to the point p_i than to any other point in P . Figure 3-8 is a Voronoi diagram.

Let $P = \{p_1, \dots, p_n\}$, where $2 \leq n < \infty$, and $x_i \neq x_j$ for $i \neq j$, $i, j \in I_n$. The function $\omega(p_i)$ is the polygonal area associated to the point p_i , and is defined by

$$\omega(p_i) = \left\{ x \mid \|x - x_i\| \leq \|x - x_j\| \text{ for } j \neq i, i, j \in I_n \right\} \quad (9)$$

The Voronoi diagram of P is $\Omega = \{\omega(p_1), \dots, \omega(p_n)\}$.

The Voronoi polygon of a point can be constructed by clipping the area with lines that bisect, and are perpendicular to, the corresponding lines between this point and its surrounding points.

In mosaic construction a simple procedure to render the mosaic is to copy, from each image, only the pixels which are closer to its centre than to any other image centre. This scheme corresponds to the Voronoi diagram. If each point corresponds to the centre of an image, the Voronoi diagram represents the portions of the images that contribute to the mosaic. The edges of the images can also be derived from the diagram for use in the blending of the images.

Delaunay triangulation is closely related to the Voronoi diagram. This triangulation is named after B. Delaunay, who first made use of this dual relationship [Delaunay, 1934]. If the Voronoi diagram is used as a basis, the Delaunay triangulation can be constructed by drawing the lines between the points in adjacent polygons i.e. those which share a common edge. The result is a triangular network. Figure 3-9 shows the Voronoi diagram of a set of points and the Delaunay triangulation superimposed.

If a mosaic is constructed using the areas of the images indicated by the Voronoi diagram, then, the points joined by a line in the Delaunay triangulation represent *neighbour images*. Neighbour images share a boundary in the mosaic, and if any

misalignment or intensity difference exists, it will show at the seams between neighbour images.

The polygonal areas given by the Voronoi diagram are not necessarily covered by the images; the polygons may extend to infinity while the image size is in fact bounded. Note in Figure 3-8 that, disregarding the clipping of the figure boundaries, the polygons on the boundary of the figure are open, i.e. they extend to infinity, because they have no neighbouring points in that direction. In addition, since the images are limited in size, not all connected points in the Delaunay triangulation will represent neighbour images. For instance, in Figure 3-9, images 1, 2, 3 and 4 are neighbours of image 17, however, image 1 is too far from image 17 to overlap. This can be resolved, when constructing the Voronoi polygons, by setting the initial area of each polygon to be equal to the image dimensions before starting the clipping. This is shown in Figure 3-10, where images 1, 2, and 3 are no longer neighbours of image 17. Figure 3-11 shows a selected image area (in blue) and its neighbour images.

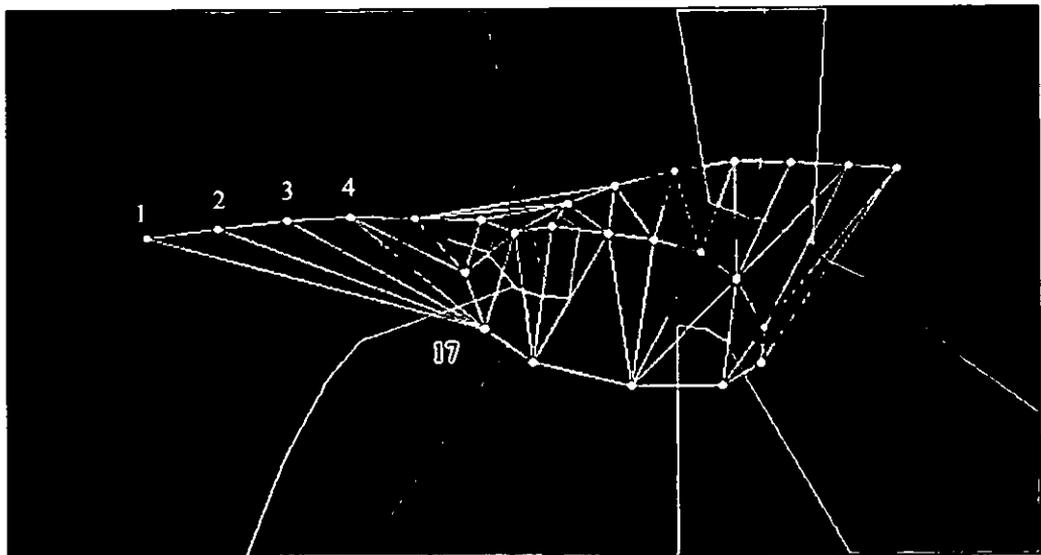


Figure 3-9 Voronoi diagram of a set of points and the Delaunay triangulation superimposed in yellow.

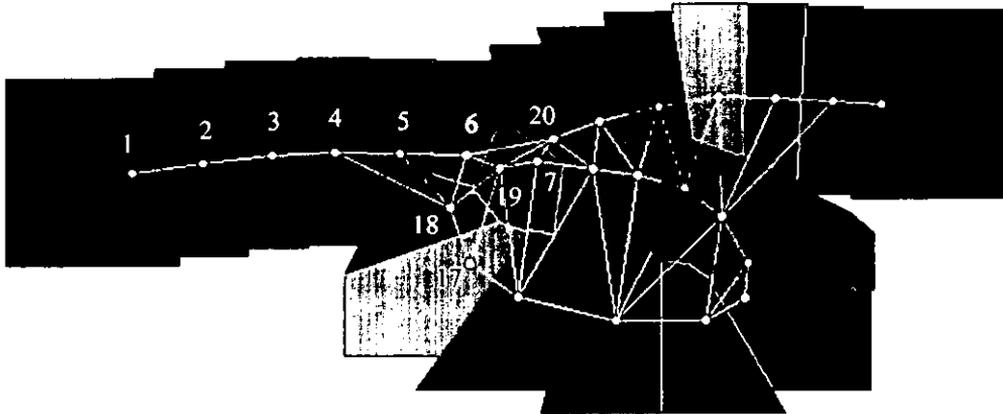


Figure 3-10 Neighbour images from Delaunay triangulation. Note points 1 and 17 are not neighbours since the images they represent do not overlap.

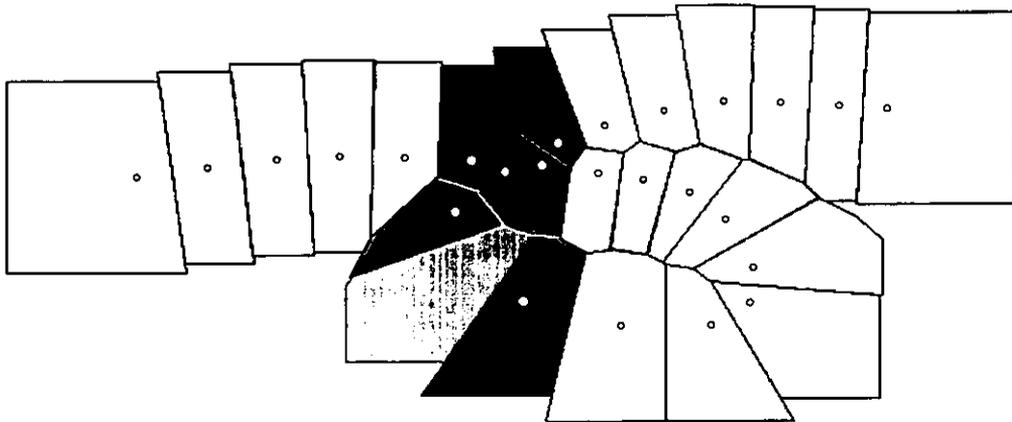


Figure 3-11 The area covered by an image (in blue), and all its neighbour images (in other colours).

3.5 Summary

This chapter has covered the techniques for image registration and image integration used in mosaic construction. It also describes the adverse effects caused by motion parallax, distortions, and moving objects. The next chapter will introduce a problem that originates from the accumulation of small alignment error over a number of images.

Chapter 4 THE LOOPING PATH PROBLEM AND PROPOSED SOLUTION

4.1 Introduction

In video sequences, successive frames are guaranteed to overlap and, therefore, in mosaicing from video sequences, the registration is computed for successive images. However, a small accumulation of alignment errors occurs when consecutive images are aligned, and in the case of the image path returning to a previous position in the mosaic a significant mismatch between non-consecutive images will result, this is referred to as the *Looping Path Problem (LPP)* in this thesis. A small alignment error will exist regardless of the image registration method used and the accuracy achieved. The effects of the looping path problem are dramatic when large numbers of images are involved in the loop, and therefore, a large accumulated error occurs (as shown in Figure 4-1).

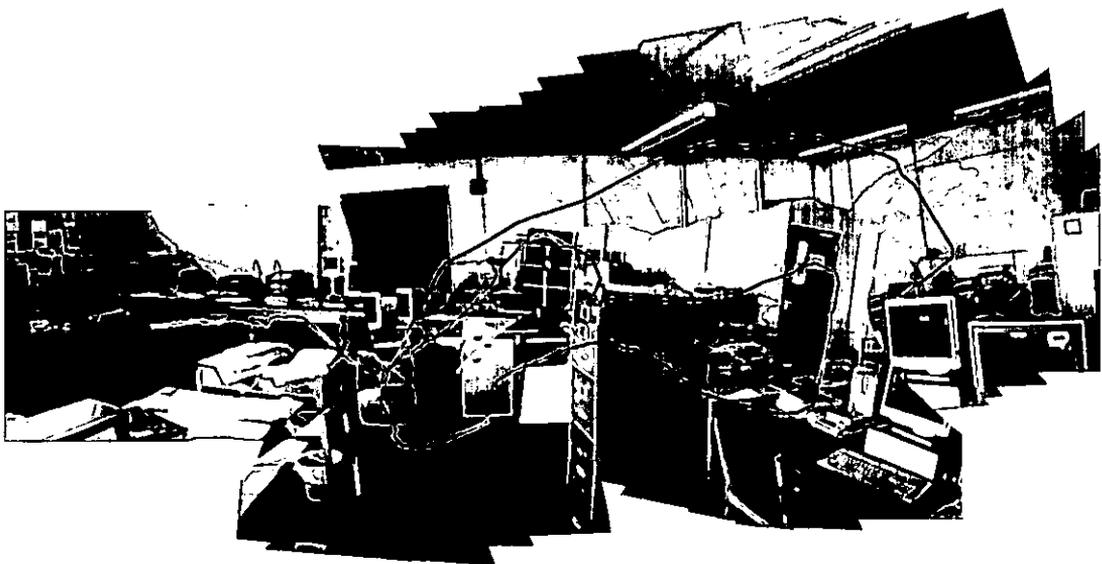


Figure 4-1 A looping path (in red) exists in this mosaic. The mismatch between different images in the region of the TV set clearly shows the need for consistency between individual alignments.

Figure 4-2 illustrates the problem for the simple case of transformations involving only translations. Assuming a perfect loop has been followed by the camera, images 1 and 60 should overlap entirely, but misalignment occurs due to accumulation of small errors in each successive image alignment. Therefore, the alignment between images 1 and 60 in Figure 4.2 is inconsistent with the position of the rest of the images in the mosaic.

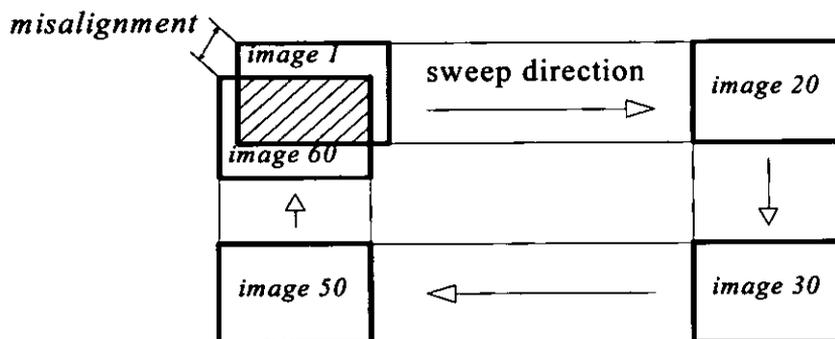


Figure 4-2 Misalignment between image 1 and image 60 due to accumulation of small errors in successive image alignment.

A new method for ensuring the consistency of the positions of all images in the mosaic is proposed in this thesis, resulting in general improvement of mosaic quality and making it possible to create very large field of view mosaics, including full spherical mosaics.

4.2 Previous Approaches

In constructing mosaics from video sequences, almost all existing methods have used parameters computed by successive image alignment. Cumulative alignment errors occur when the position of images in the mosaic is based on successive image alignment only. Good alignment is typically achieved for each individual image alignment, however, cumulative errors cause poor alignment when the image path follows a loop, i.e. when the same area of the scene is covered by images which are distant in the sequence.

Where most of the proposed methods to reduce the LPP are designed for real time mosaic construction (where the mosaic has to be updated continually with every new

incoming image without much reference to distant images in the sequence), these are not adequate for high quality imaging.

This problem has been identified in literature and several solutions proposed. These approaches are discussed in the following sections.

4.2.1 Registering New Image with Current Mosaic

In looking for a solution to the looping path problem it has been proposed that rather than aligning successive images, the alignment can be done instead between an image and the actual mosaic as it is being composited [Burt, Hansen and Anandan, 1996] [Irani, Anandan and Hsu, 1996] [Sawhney and Kumar, 1997]. This solution can be implemented for real-time video mosaicing, that is, when all the images are not available simultaneously, but arrive periodically as a stream of images that need to be integrated into the present mosaic.

This represents an improvement with respect to the successive image alignment method, however, loops involving large numbers of images result in distortions in the mosaic. In some cases, the next image to be aligned with the current mosaic will need to fit two or more different transformations, and distortion will be inevitable. For instance, image 60 in Figure 4-2 will be distorted if an attempt is made to align it simultaneously with image 59 and with image 1.

4.2.2 Registering Sub-mosaics

A similar solution was addressed by Mann and Picard [Mann and Picard, 1995]. They suggested that the sequence of images be split into subsets which are well registered. This produces *sub-mosaics* which are then globally registered to form the final mosaic. This technique is fast and fairly straightforward to implement, but deciding how to subdivide the image sequence can be problematic, hence a distortion-free mosaic is not ensured.

4.2.3 Distribution of the Error

In constructing panoramic (i.e. cylindrical) mosaics, from a series of images acquired by a panning camera (as shown in Figure 3-5), the looping path problem is reduced to

solving how to match the first image with the last one while keeping the consistency with the rest of the alignments.

[Szeliski, 1994b] proposed a method to ‘close the gap’ in a panorama. He solves this problem by registering the same image at the beginning and at the end of the sequence. The error can be distributed evenly across the whole sequence. This technique updates the focal length and transformation matrices so that the sum of registration errors between all matching pairs of images is minimised [Szeliski and Heung-Yeung, 1997]. A similar approach is found in [Fitzgibbon and Zisserman, 1998] which also distributes the error in closed image sequences.

4.2.4 Simultaneous Registration of All Images

Recently, [Capel and Zisserman, 1998] have developed a technique to consistently align the images in a mosaic. Their technique uses Harris corners as salient features for image registration. After identifying a minimum of four common corners per pair of overlapping images, the projective transformation is determined. A cost function is then defined based on the Euclidean distance between corresponding points. A maximum likelihood estimator, generalised to handle point matches over many images, is used to estimate a set of consistent transformations.

This elegant approach effectively solves the looping path problem. Yet, the computational cost and the limitations derived from feature correspondence can be significant. The computational cost of this approach can be considerable due to the very large number of parameters, especially when the number of images involved in the alignment process is large.

This technique is essentially one of image registration, extended to compute simultaneously all the transformations for all pairs of overlapping images. Registration techniques using feature points (corners in this case) rely on the presence of such features in the scene. This is a limiting factor for the content of the images. Scenes with too many corners (e.g. repetitive patterns) or too few corners yield to problems when matching too many points or an insufficient number of points.

Although other feature points can be used instead of corners, this method inherits the restrictions associated with the use of feature matching. Due to computational limitations, it cannot be realistically converted for use with other registration techniques such as direct estimation (as discussed in section 3.2.3), which use all the data and structure constraints. Nevertheless, it should be noted that robustness can be added to this approach by combining feature point matching with direct estimation or another robust registration technique.

4.3 Proposed Solution to the Looping Path Problem

A solution that ensures the consistency of the positions of all the images in a mosaic is introduced in this section. The proposed solution minimises the error of the relative positions of all pairs of neighbour images in the mosaic. Any image registration technique can be used to align the images, and this solution does not impose any restriction on the reprojection surface, allowing full spherical mosaics to be created.

The process for consistent image alignment is summarised in Figure 4-3.

```

1.  Compute alignments of successive images
2.  Create list of neighbour images
3.  WHILE (new neighbours arise) {
4.      Compute alignments of new neighbour images
5.      Minimise error by modifying positions of images
6.      Update list of neighbour images
    }

```

Figure 4-3 Pseudocode of process for consistent image alignment.

Line 1: Compute Alignments of Successive Pairs Of Images.

The process begins by computing the alignment of successive images to obtain an initial position of the images on the mosaic. Successive images in video sequences overlap significantly, and hence abundant information, necessary for the registration, is available. The registration which is found to be at this stage is typically accurate to 1/10 of a pixel.

Neighbour images are those that share a boundary in the mosaic (as explained in section 3.4). If a pair of neighbour images is represented by $\mathbf{N}_k = \{i, j\}$ where image i and image j are neighbour images, the list of pairs of neighbour images can be initialised with successive image pairs as follows

$$\mathbf{N}_k = \{k, k+1\}, \text{ for } 0 \leq k < N-1 \quad (10)$$

where N is the number of images.

Line 2: Create List of Neighbour Images.

The positions of the images on the mosaic can be computed using the successive alignments by concatenation of transformations. Although the alignment error is accumulated at this stage, these positions are sufficiently correct to reveal new neighbour images. This second line updates the list of neighbour images according to the Delaunay triangulation explained in section 3.4.

There are cases where the path crosses itself and inserts a new image between two consecutive images. For instance, in Figure 3-10, images 6 and 7 are not neighbours since image 19 falls in between them. Since misalignment inaccuracies are only visible at the seams between neighbour images, and there is one seam per pair of neighbour images, only these alignments should be considered. That is, when the list of neighbour images is updated, pairs of consecutive images may have to be deleted from the list.

Line 3: WHILE (New Neighbours Arise)

If a looping path exists, the list of neighbour images will include alignments not yet computed between non-successive images. If no looping path exists, no new neighbours will arise, and therefore the process would be completed.

Line 4: Compute Alignment of New Neighbour Images.

If, however, a looping path is followed by the images, the alignments of the new neighbour images will be computed.

Line 5: Minimise Error Function by Modifying Positions of Images.

A cost function will be defined that represents the positional error between two neighbour images. This error is a measure of the difference between the computed alignment and the actual position on the mosaic.

Figure 4-4 shows a simple case of a mosaic constructed from images related by only translations, where $\{p_0, p_1, p_2, \dots\}$ are the position of the centres of the images on the mosaic, and $\{t_{01}, t_{12}, t_{23}, \dots\}$ are the alignments computed by registration of successive images. If images 0 and 4 are neighbours, and t_{04} is the registration that aligns them, then the error E_{04} is the difference between the relative position given by the registration and the actual relative position in the mosaic:

$$E_{04} = t_{04} - (p_4 - p_0) \quad (11)$$

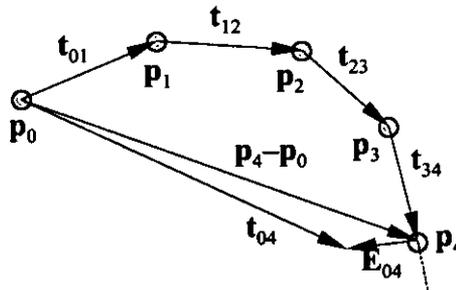


Figure 4-4 The positional error between images 0 and 4 is given by E_{04}

Initially, for successive images (i.e. $j = i+1$), $E_{ij} = 0$. However, for the rest of the neighbour images (i.e. $j \neq i+1$) $E_{ij} \neq 0$ due to the accumulated error in the successive alignments. E_{ij} is the error to be distributed for all i, j neighbour images.

The relative position of a pair of neighbour images on the mosaic can be modified slightly without introducing a visible loss in quality. Such a change from its computed registration must not exceed a fraction of a pixel if the seam is to remain unnoticeable. Therefore, the summation of the errors from all individual pairs of neighbour images can be minimised by modifying the positions of the images on the mosaic, hence reducing the gross visible error at the seams in a looping path.

The minimisation problem is implemented using an analogy with a physical model. Each image can be modelled as a rigid body on which forces are exerted. A rigid body has a spatial position and orientation. Forces can be modelled to modify the position and orientation of the images according to neighbour relationships. A physical simulation can provide an efficient way of specifying these relationships. This solution to the minimisation problem will be described in detail in the next sections.

Line 6: Update List of Neighbour Images

After the positions of the images have been rearranged to minimise the error, new neighbours may arise and will be added to the list, and other pairs of images may no longer share a boundary and will be deleted from the list. The list of neighbour images will be updated, and, as long as new neighbours appear, the process will carry on iteratively as specified in line 3.

4.4 Minimisation using Physical Simulation

In solving the LPP, line 5 in Figure 4-3 is a minimisation problem in which the cost function depends on the positions of the images on the mosaic and their relative computed alignments. This can be implemented using an analogy with a physical model. In this model, each image is represented by a rigid body where forces are exerted to attract them to the correct position with respect to each other.

4.4.1 Why Physical Simulation?

A physical model can be devised by modelling one image and its behaviour with respect to other images, then a system with any number of images can be simulated, and their evolution towards the desired consistent positions monitored for optimisation.

An analogy with a physical model has been chosen to minimise the cost function for a number of reasons, these include visualisation, modularity and interactivity.

A physical simulation permits the *visualisation* of the model as a set of images in different positions and orientations in space. The topology of the set of images as they move towards equilibrium can be monitored and provides important feedback on the suitability of the model used. Although the evolution of the positions of the images is simple in the case of a two-dimensional mosaic where the images move within a plane, it is polymorphic in the case of the cylindrical and spherical mosaic in three-dimensional space. Examples of these benefits are given in section 6.4.2.

The *modularity* allows the modelling of complex relationships by using simple elements such as springs and dampers. Auxiliary forces can be introduced logically into the model to guide the images to the stable equilibrium and to introduce the constraints needed for particular cases such as the spherical distribution of images.

Interaction with the model, in the form of adding and removing constraints, such as the constraints for the spherical mosaic, is necessary in order to gain a comprehensive understanding of the effect of such constraints on convergence and stability.

The simulation also provides a means of assessing the convergence to equilibrium by examining its energy. When the system reaches the rest state, its total energy (that is, the sum of kinetic and potential energy) is minimum and its kinetic energy is zero. If different equilibrium states are reached, they can be compared in terms of the remaining potential energy, which indicates the overall degree of alignment achieved in each case.

Hence, the physical simulation helps to identify and model the forces necessary for the consistent alignment of images.

4.4.2 Physical Simulation

A *physical simulation* is the computation of the equations of motion of a physical system, consisting of one or more bodies, according to the forces exerted upon them. The variables typically investigated in a physical simulation include acceleration, velocity, and the spatial variables, i.e. position and orientation. These are given initial values, then the simulation is run. Depending on the purpose of the simulation, the focus is set on the spatial variables, the velocities and accelerations, or on the stability of the simulated system. Variable values can be monitored during a simulation, and they are of great value in system design.

4.4.2.1 The Rigid Body

A *particle* is an idealised entity with no size or internal structure. A particle is defined by its mass and its position in space, hence, all the mass of a particle is supposed to be concentrated at its position. In the real world we must deal with extended objects instead of particles. Some extended objects, such as a pencil eraser or a spring, are flexible and can change their shape or size in response to applied forces. On the other hand, a mechanic's wrench remains rigid and nondeformable under ordinary circumstances. A *rigid body* is one for which the distance between any pair of points on the object remains fixed.

A particle is characterised by its mass, position and velocity in space. Additionally, rigid bodies possess orientation in space and hence angular velocity. Position and orientation are defined with respect to a reference coordinate system.

4.4.2.2 Coordinate Systems

The spatial variables of a rigid body define its position and orientation with respect to a *reference coordinate system* or *reference frame*. This is specified by three orthogonal unitary vectors \mathbf{x} , \mathbf{y} , and \mathbf{z} .

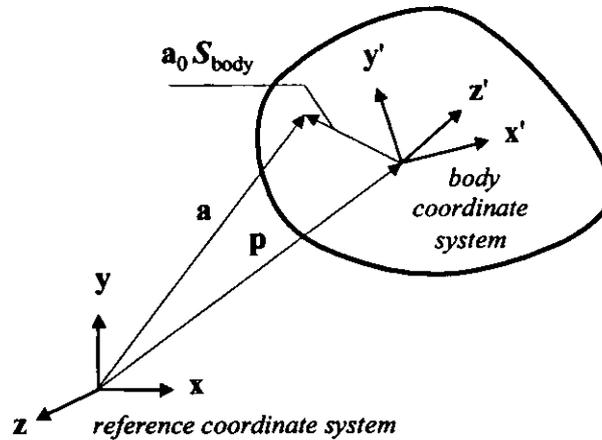


Figure 4-5 Position and orientation of a rigid body with respect to a reference coordinate system.

The position \mathbf{p} of a rigid body can be described as its translation from the origin of the reference coordinate system as shown in Figure 4-5. The components of the vector \mathbf{p} in world-space coordinates are p_x , p_y and p_z , such that

$$\mathbf{p} = p_x \mathbf{x} + p_y \mathbf{y} + p_z \mathbf{z} \quad (12)$$

The components of a vector are said to be given in *world-space coordinates* if they are referred to the reference coordinate system.

The orientation in space of a rigid body can be specified by a coordinate system attached to the body. The *body coordinate system* or *body frame* is defined by three orthogonal unitary vectors namely \mathbf{x}' , \mathbf{y}' , and \mathbf{z}' . This can be expressed in matrix form as a 3×3 matrix where each vector is a row.

$$\mathbf{S}_{\text{body}} = \begin{pmatrix} \mathbf{x}' \\ \mathbf{y}' \\ \mathbf{z}' \end{pmatrix} = \begin{pmatrix} x'_x & x'_y & x'_z \\ y'_x & y'_y & y'_z \\ z'_x & z'_y & z'_z \end{pmatrix} \quad (13)$$

The position \mathbf{p} and the orientation \mathbf{S}_{body} are called the *spatial variables* of a rigid body.

If a vector is given in *body-space coordinates*, its components are referred to the body coordinate system. A point given in body-space coordinates can be expressed in world-space coordinates. Let \mathbf{a}_0 be an arbitrary point on the rigid body in body-space coordinates, then point \mathbf{a} in world-space coordinates is given by

$$\mathbf{a} = \mathbf{a}_0 \mathbf{S}_{\text{body}} + \mathbf{p} \quad (14)$$

4.4.2.3 Linear Equations of Motion

The equations of motion provide the trajectory of a body and its linear and angular accelerations and velocities, according to the forces acting on it. In this and the following sections, the basic concepts of dynamics are reviewed.

Newton's second law [Newton, 1686] is concerned about the change in velocity that results from the application of a force. Newton's second law states, *if a net force is acting on an object, it will accelerate in the direction of the force, and the magnitude of the acceleration is proportional to the magnitude of the force*. The second law is often written in the following mathematical form:

$$\sum_i \mathbf{F}_i = m\mathbf{a} \quad (15)$$

where $\Sigma \mathbf{F}_i$ is the resultant sum of all the forces exerted on the object,
 \mathbf{a} is the *acceleration*, and
 m represents the mass of the object.

The *velocity* \mathbf{v} of an object varies according to its acceleration \mathbf{a} (i.e. the rate of change of velocity). Similarly, the *position* \mathbf{p} of an object varies with its velocity \mathbf{v} (velocity is the rate of change of position).

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} \qquad \mathbf{v} = \frac{d\mathbf{p}}{dt} \quad (16)$$

4.4.2.4 Angular Equations of Motion

Newton's second law in angular form is

$$\sum_i \tau_i = I \alpha \quad (17)$$

where τ_i is the torque generated by the force F_i acting on the rigid body,
 I is the rotational inertia matrix of the body, and
 α is the resulting *angular acceleration* about that axis.

Torque ($\vec{\tau}$) is the turning or twisting action on a body about a rotation axis due to a force F . If F is exerted at a point given by the position vector r relative to the axis, then the torque τ is the cross product of r_i and F .

$$\tau_i = r_i \times F_i \quad (18)$$

The rotational *inertia* matrix I of a body describes how the mass of a body is distributed relative to the body's centre of mass. The inertia depends on the orientation of a body, but does not depend on its translation. At any given time t , let $r_i = (r_{ix} \ r_{iy} \ r_{iz})$ be the displacement in world-space coordinates, with respect to the centre of mass of the body, of the i^{th} particle of mass m_i . The inertia is expressed in terms of a symmetric matrix.

$$I(t) = \sum m_i \begin{pmatrix} r_{iy}^2 + r_{iz}^2 & -r_{ix}r_{iy} & -r_{ix}r_{iz} \\ -r_{iy}r_{ix} & r_{ix}^2 + r_{iz}^2 & -r_{iy}r_{iz} \\ -r_{iz}r_{ix} & -r_{iz}r_{iy} & r_{ix}^2 + r_{iy}^2 \end{pmatrix} \quad (19)$$

For a body with continuous distributed mass, the finite sums can be replaced with integrals over its volume.

Whenever orientation changes, the inertia matrix changes. However, by using a precomputed inertia matrix in body-space coordinates, the inertia matrix can be effectively computed for any orientation $S(t)$ [Baraff and Witkin, 1997].

$$I(t) = S_{\text{body}(t)}^T I_{\text{body}} S_{\text{body}(t)} \quad (20)$$

The inverse of $I(t)$ is also used in a physical simulation, and it is given by the equation

$$I(t)^{-1} = S_{\text{body}(t)} I_{\text{body}}^{-1} S_{\text{body}(t)}^T \quad (21)$$

The *angular velocity* ω of a rigid body varies according to its angular acceleration α , which is the rate of change of angular velocity. Finally, the *angular position* of a rigid body varies with its angular velocity.

$$\alpha = \frac{d\omega}{dt} \qquad \omega = \frac{d\theta}{dt} \quad (22)$$

4.4.2.5 Simulation Engine

A simulation engine solves numerically the equations of motion reviewed in the sections 4.4.2.3 and 4.4.2.4. These are differential equations for which symbolic solutions (i.e. those in which the form of the unknown function is to be guessed) can only be implemented for the simplest types of motion. In contrast, numerical solutions are specifically suited to computer implementation regardless of the complexity of the equations. In numerical solutions discrete time steps are taken, after initialising the variables of motion, to compute iteratively the values in time of these variables. In each iteration, the motion parameters for each body contained in the simulated world are evaluated for a time step Δt .

The forces acting on the body and the current state of the body (i.e. its position, orientation and velocities at a time t) are the input values in each iteration. The result is the new values of these variables at a time $t+\Delta t$.

The problem can be summarised as obtaining the values of

$$\mathbf{a}(t+\Delta t), \mathbf{v}(t+\Delta t), \mathbf{p}(t+\Delta t), \alpha(t+\Delta t), \omega(t+\Delta t) \text{ and } S_{\text{body}(t+\Delta t)} \quad (23)$$

given

$$\{\mathbf{F}_1(t), \dots, \mathbf{F}_n(t)\}, \mathbf{v}(t), \mathbf{p}(t), \boldsymbol{\omega}(t), \mathbf{S}_{\text{body}}(t), m, \mathbf{I}, \text{ and } \Delta t \quad (24)$$

where $\{\mathbf{F}_1(t), \dots, \mathbf{F}_n(t)\}$ are the forces acting on the body.

4.4.2.6 Forces

The forces involved in a physical simulation are normally known (e.g. weight) or can be calculated from known parameters such as constants, distances (e.g. springs, gravitational forces), velocities (e.g. dampers, aerodynamic drag), etc.

Most of the equations that model the forces existing in nature have been found by observation, and these can achieve different degrees of accuracy in the prediction of the correct value in the real world. For example the lift force acting on an aircraft wing can be approximately modelled as $F_{\text{lift}} = \frac{1}{2} \rho S V^2 C_l$ which depends on the density of the air (ρ), the area of the wing (S), the velocity of the air (V), and an empirical lift coefficient (C_l) which is dependent on the angle of attack.

Forces are vectors expressed in world-space coordinates. The direction and magnitude of the vector indicates the direction and magnitude of the force. Additionally, each force \mathbf{F}_i has associated a point of application \mathbf{r}_i used in the computation of the torques.

4.4.2.7 Euler's Integration Method

The simplest numerical method to solve the differential equations of motion is called Euler's integration method.

To obtain the values of the variables of motion at a time $t + \Delta t$, knowing their values at a time t , the following steps can be taken:

Linear Variables

The variables to be initialised for the motion of a body are \mathbf{v}_0 , \mathbf{p}_0 , $\boldsymbol{\omega}_0$ and $\mathbf{S}_{\text{body}0}$.

Linear Acceleration

The linear acceleration can be computed from the forces acting on the body and its mass as follows

$$\mathbf{a}(t) = \frac{\sum_{i=1}^n \mathbf{F}_i(t)}{m} \quad (25)$$

where n is the number of forces acting on the body.

Linear Velocity

Acceleration is the rate of change of velocity, hence the variation $\Delta \mathbf{v}$ of the velocity in a time step Δt is

$$\Delta \mathbf{v} = \mathbf{a}(t) \Delta t \quad (26)$$

therefore, the value of velocity at $t+\Delta t$ is

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \Delta \mathbf{v} \quad (27)$$

Position

Analogously, the velocity (which is the rate of change of position) is used to update the position of the body. A time step Δt is taken which gives a variation of position $\Delta \mathbf{p}$. This variation is used to obtain the position of the body at $t+\Delta t$

$$\Delta \mathbf{p} = \mathbf{v}(t+\Delta t) \Delta t \quad \mathbf{p}(t+\Delta t) = \mathbf{p}(t) + \Delta \mathbf{p} \quad (28)$$

Angular Variables

The angular equations of motion show a parallelism with the linear equations of motion, where *mass* is substituted by *inertia* and *force* by *torque*.

Torque

The net torque $\tau(t)$ acting on a body is calculated from the forces and their points of action, using (18) as follows

$$\tau(t) = \sum_{i=1}^n \tau_i(t) \qquad \tau_i(t) = \mathbf{r}_i \times \mathbf{F}_i(t) \qquad (29)$$

The point of application \mathbf{r}_i must be expressed in world coordinates, and, as in equation (14), it is given by

$$\mathbf{r}_i = \mathbf{r}_{0i} \mathbf{S}_{\text{body}} \qquad (30)$$

where \mathbf{r}_{0i} is the point of application of the force in body space coordinates.

Angular Acceleration

From equation (17), the angular acceleration can be expressed as

$$\alpha(t) = \mathbf{I}(t)^{-1} \tau(t) \qquad (31)$$

where the inertia matrix $\mathbf{I}(t)$ has been explained in section 4.4.2.4, and its inverse can be calculated using equations (19), (20) and (21).

Angular Velocity

As with the linear velocity, the angular velocity at $t+\Delta t$ is calculated by adding to the angular velocity at t , the variation of angular velocity $\Delta\omega$

$$\Delta\omega = \alpha(t) \Delta t \qquad \omega(t+\Delta t) = \omega(t) + \Delta\omega \qquad (32)$$

Orientation

Finally, the orientation matrix \mathbf{S}_{body} must be updated according to the angular velocity. The increment in angular velocity represents the rotation increment that corresponds to Δt . The rotation of the body is accomplished by rotating the three axes that define the body coordinate system about $\Delta\omega$.

$$\mathbf{S}_{\text{body}}(t+\Delta t) = \mathbf{R} \mathbf{S}_{\text{body}}(t) \quad (33)$$

where \mathbf{R} is a rotation matrix for which an explanation can be found in [Faux and Chichester, 1979]. This matrix rotates a point by an angle ω about an axis given by a unitary vector \mathbf{u}

$$\mathbf{R} = \begin{pmatrix} u_x^2 + \cos(1 - u_x^2) & u_x u_y (1 - \cos \omega) - u_z \sin \omega & u_z u_x (1 - \cos \omega) + u_y \sin \omega \\ u_x u_y (1 - \cos \omega) + u_z \sin \omega & u_y^2 + \cos(1 - u_y^2) & u_y u_z (1 - \cos \omega) - u_x \sin \omega \\ u_z u_x (1 - \cos \omega) - u_y \sin \omega & u_y u_z (1 - \cos \omega) + u_x \sin \omega & u_z^2 + \cos(1 - u_z^2) \end{pmatrix} \quad (34)$$

The direction and magnitude of the angular velocity vector gives the axis \mathbf{u} and the angle of rotation $\Delta\omega$.

$$\mathbf{u} = \frac{\Delta\boldsymbol{\omega}}{\|\Delta\boldsymbol{\omega}\|} \quad \Delta\omega = \|\Delta\boldsymbol{\omega}\| \quad (35)$$

This method for updating the orientation yields numerical drift due to limited floating point accuracy, and the three unitary vectors defining the body coordinate system may no longer be unitary nor orthogonal. Graphically, the effect would be that applying \mathbf{S}_{body} to a body would cause a skewing effect. This problem can be alleviated by normalising and forcing orthogonality in the orientation matrix \mathbf{S}_{body} . Another way of representing the orientation of a rigid body than using a 3×3 matrix is with unit quaternions. Quaternions are a type of four element vector normalised to unit length and since they have only four parameters to represent the three freedoms of the rotation the drift is less. Information on the use of quaternions can be found in [Baraff and Witkin, 1997].

4.4.2.8 Midpoint and Runge-Kutta Integration Methods

Though simple, Euler's method is not accurate. Consider the case of a 2D function f whose integral curves are concentric circles. A point \mathbf{p} governed by f is supposed to orbit forever on whichever circle it started on. Instead, with each Euler step, \mathbf{p} will move on a straight line to a circle of larger radius, so that its path will follow an outward spiral. Reducing the step size will slow the rate of this outward drift, but never eliminate it.

Euler's method can also be unstable leading to divergent solutions. Finally, Euler's method is not efficient, since to achieve a reasonable accuracy and stability, the step size must be kept small. Other common integration methods such as *midpoint* or *Runge-Kutta* can be used which improve accuracy, stability and efficiency.

The *midpoint* method is an improvement over Euler's method. It is a 2nd order solution method which computes a Euler step (36) at an initial point $x=o$ (see Figure 4-6). This step takes the point to the position a . Then the derivative is evaluated again (37) at step's midpoint m (which takes the point from m to b), and the second evaluation is used to calculate (38) the actual step to take (from o to c).

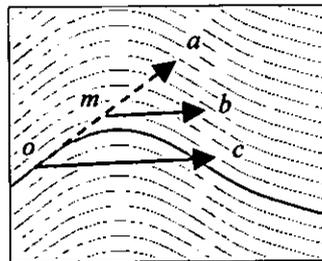


Figure 4-6 Midpoint integration.

$$\Delta x = \Delta t f(x, t) \quad (36)$$

$$f_m = f\left(x + \frac{\Delta x}{2}, t + \frac{\Delta t}{2}\right) \quad (37)$$

$$x(t+\Delta t) = x(t) + \Delta t f_m \quad (38)$$

Runge-Kutta is an order 4 method which further reduces the error by evaluating 4 times the derivative at different step points before taking a step (a more comprehensive explanation on midpoint and Runge-Kutta methods can be found in [Press et al., 1992]).

Determining the step size is a major problem in any of the methods introduced. Ideally Δt should be as large as possible, but not so large as to cause an unreasonable amount of error, or worse still, to induce instability. With a fixed step size, it can only be proceeded as fast as the “worst” sections of the function to integrate allows. Adaptive stepsizing is a technique that modifies the step size to be as large as possible, and at the same time sufficiently small to avoid excessive error.

The accuracy of an integration method refers to the capability to compute accurate values of the variables of motion at any given time. For the purpose of consistently aligning the images, only the final position of the images are of interest, that is, at the time when equilibrium or rest is achieved. Since the increments of the variables of motion (Δp , Δv , etc.) tend to zero as equilibrium is approached, Euler’s method is as accurate as the other methods mentioned in this section at the final equilibrium position.

Both Euler and midpoint integration methods have been implemented for this project, giving similar transient results and identical end positions for the images. However, the Euler method needed smaller time steps than the midpoint algorithm to prevent unwanted oscillations.

4.4.3 Physical Model for the Images

Each image is modelled as a rigid body with unitary mass and unitary moment of inertia, the centre of mass being situated at the centre of the image (see Figure 4-7).

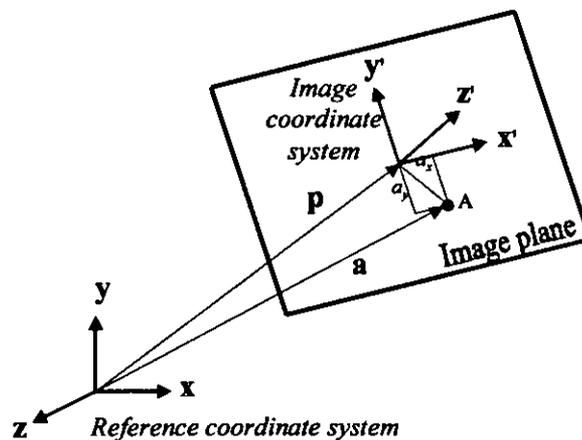


Figure 4-7 Position and orientation of an image as a rigid body.

The position of an image in space is described as a vector \mathbf{p} , which is the translation of its centre from the reference coordinate system's origin.

The three unitary orthogonal vectors that define the image coordinate system are \mathbf{x}' , \mathbf{y}' and \mathbf{z}' . The vector axis \mathbf{z}' is perpendicular to the image plane, and \mathbf{x}' and \mathbf{y}' lie on the image plane and define the conventional horizontal and vertical axes used to locate the pixels on an image.

A vector is said to be given in *image space* coordinates if its components are referred to the image coordinate system.

The position of a point \mathbf{A} which lies on the image plane at image coordinates $\mathbf{a}_0 = (a_x, a_y, 0)$ is expressed in world coordinates as

$$\mathbf{a} = \mathbf{a}_0 \mathbf{S}_{\text{image}} + \mathbf{p} \quad \text{or} \quad \mathbf{a} = \mathbf{p} + a_x \mathbf{x}' + a_y \mathbf{y}' \quad (39)$$

4.4.4 The Hinge Constraint

Two images recorded by a rotating ideal camera (i.e. as described in section 3.2.2, Figure 3-6) are related by a projective transformation. This is actually the transformation that relates the camera's sensor plane positions and orientations in space at the time the images were captured. Figure 4-8 shows the images projected onto the sensor plane of a camera as the camera rotates about its optical centre.

The intersection of the sensor planes at these two different times is a line, and through that line the intensity values for both images coincide. Therefore, when the position in space of two images is such that they cross through the *intersection line*, no misalignment is shown. In addition, if the images are viewed from the intersection point of the perpendiculars to the images passing through their respective centres, which corresponds to the optical centre, there will be no projective distortion [Gümüstekin and Hall, 1996b].

The *intersection line* can be extracted by image registration. If the rotation angle of the camera is small the rigid model can be used without introducing significant distortion [Mann and Picard, 1994c] [Mostafavi and Smith, 1978]. However, the angle between the image planes is unknown if only the intersection line is known. The angle depends on the field of view of the camera (or equivalently, its focal length), which is needed to construct the mosaic.

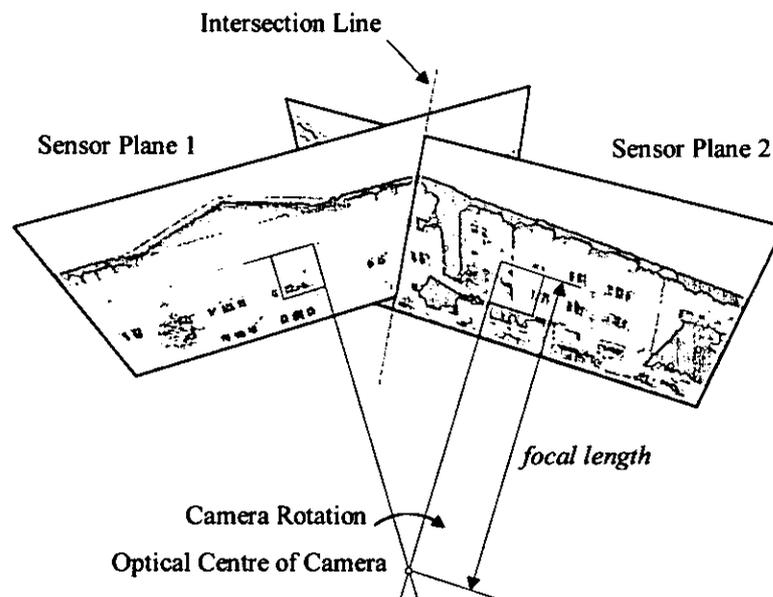


Figure 4-8 The position and orientation of the sensor plane at the time the images were captured are closely related to the position and orientation of the images.

Based on the relationship between the sensor plane and the image position, a constraint can be used to impose this relative position on a pair of overlapping images.

The constraint designed for this research project is a *non-rigid hinge*, defined, for every pair of overlapping images A and B, by two points, P_{H1A} and P_{H2A} , in image A which are attracted to the corresponding two points, P_{H1B} and P_{H2B} , in image B. This is shown in Figure 4-9. The attraction is a force whose magnitude depends on the distance between corresponding points.

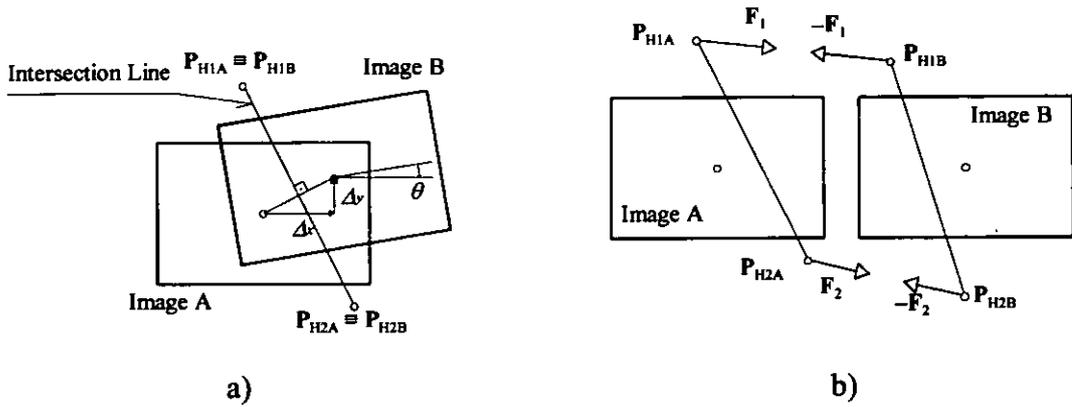


Figure 4-9 The hinge model. a) When the two images are correctly aligned, the hinge points coincide ($P_{H1A} \equiv P_{H1B}$ and $P_{H2A} \equiv P_{H2B}$). b) When the images are separated, forces F_1 and F_2 will pull the images to the correct positions.

Newton's third law states *action and reaction are equal in magnitude and opposite in direction*. Hence, if F_1 is exerted on P_{H1A} , its reaction force $-F_1$ will be exerted on P_{H1B} .

This can be implemented with forces proportional to the distance between hinge points.

$$\begin{aligned} \mathbf{F}_1 &= k_H (\mathbf{P}'_{H1B} - \mathbf{P}'_{H1A}) \\ \mathbf{F}_2 &= k_H (\mathbf{P}'_{H2B} - \mathbf{P}'_{H2A}) \end{aligned} \quad (40)$$

The hinge points are defined in image space coordinates, i.e. they are located at a fixed position relative to the image coordinate system. The separation between hinge points ($\mathbf{P}_{H1A} - \mathbf{P}_{H2A}$) has the following effect. If the hinge points are very close, the torque between the two images would be very small, and the images could easily rotate with respect to each other. If the hinge points are far apart, a high torque would keep the deviation of the angle between the images small. Thus the separation between hinge points determines the rigidity of the pair of images with respect to rotation.

For any pair i of neighbour images, the measure of the error between their current position in space and their actual registration is given by

$$E_i = \frac{|\mathbf{P}'_{H1A_i} - \mathbf{P}'_{H1B_i}| + |\mathbf{P}'_{H2A_i} - \mathbf{P}'_{H2B_i}|}{2} \quad (41)$$

which is the sum of the Euclidean distances between corresponding hinge points, where P'_{H1A_i} , P'_{H1B_i} , P'_{H2A_i} and P'_{H2B_i} are the hinge points expressed in world coordinates. This error is an indication of the image quality at the seam.

4.4.5 Stability and Convergence of Image System

A system of images linked together using the hinge constraint explained previously has a series of properties which are common to other physical systems. The simple cases of spring-mass and damped spring-mass systems are reviewed in the sections 4.4.5.1 and 4.4.5.2. From these simple cases, important conclusions can be drawn on stability and convergence speed from the parameters involved in the image system.

4.4.5.1 Spring-Mass System

Equation (40) describes a force known as Hooke's law after Robert Hooke. It models a mechanical spring where k is the proportionality constant. The force F exerted by the spring is proportional to the displacement x of the free end from its position when the spring is in the relaxed state.

$$F = -kx \quad (42)$$

The block-spring arrangement of Figure 4-10 forms a linear simple harmonic oscillator, where 'linear' indicates that force F is directly proportional to displacement x rather than some other power of x . The angular frequency ω of the simple harmonic motion of the block is related to the spring constant k and the mass m of the block.

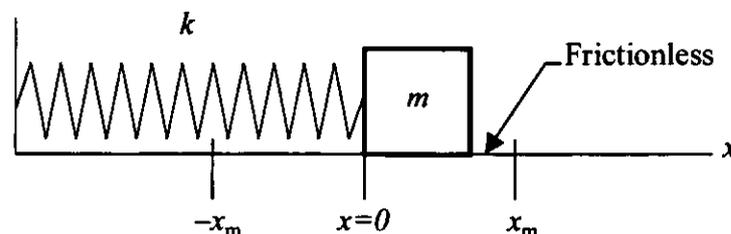


Figure 4-10 A simple harmonic oscillator. The block moves in simple harmonic motion once it has been pulled to the side and released.

In simple harmonic motion, the displacement is given by

$$x(t) = x_m \cos(\omega t + \phi) \quad (43)$$

in which x_m is the amplitude of the displacement, the quantity $(\omega t + \phi)$ is the phase of the motion and ϕ is the phase constant. The angular frequency is

$$\omega = \sqrt{\frac{k}{m}} \quad (44)$$

The potential energy associated with a spring is called the *elastic potential energy* of the spring, and is given by

$$U = \frac{1}{2} kx^2 \quad (45)$$

This potential energy is never negative because it is proportional to the square of x . If the spring is compressed, making x negative, the potential energy is again positive. The elastic potential energy is zero only if $x=0$, corresponding to a relaxed spring.

The *kinetic energy* of the mass is given by

$$K = \frac{1}{2} mv^2 \quad (46)$$

where v is the velocity of the mass.

The *mechanical energy* E is the sum of the potential energy U and the kinetic energy K .

$$E = U + K \quad (47)$$

The system is conservative since only conservative forces (elastic force is a conservative force) do work on the mass. This means that mechanical energy is conserved, that is, it does not change during the motion. A decrease in the kinetic energy of the mass is accompanied by an equal increase in potential energy. Potential energy can be thought of as being “stored energy”, since it has the “potential” of being converted into kinetic energy. Kinetic and potential energies are transformed into each other during motion such that the net mechanical energy remains unchanged.

It is evident that unless some sort of damping is introduced, the mass will oscillate indefinitely, the amplitude of the motion being constant on either side of the equilibrium position. In a system of several masses connected with springs, the motion will be oscillatory and will continue indefinitely.

4.4.5.2 Damped Spring-Mass System

A damping force (non-conservative force) proportional to the velocity and in the opposite direction is added to the spring-mass system of the previous section. An idealised example of a damped oscillator is shown in Figure 4-11: a block with mass m oscillates on a spring with spring constant k . From the mass, a rod extends to a vane (both assumed massless) that is submerged in a liquid. As the vane moves up and down, the liquid exerts an inhibiting drag force on it and thus on the entire oscillating system. With time, the mechanical energy of the block-spring system decreases, as energy is transferred to thermal energy of the liquid and vane.

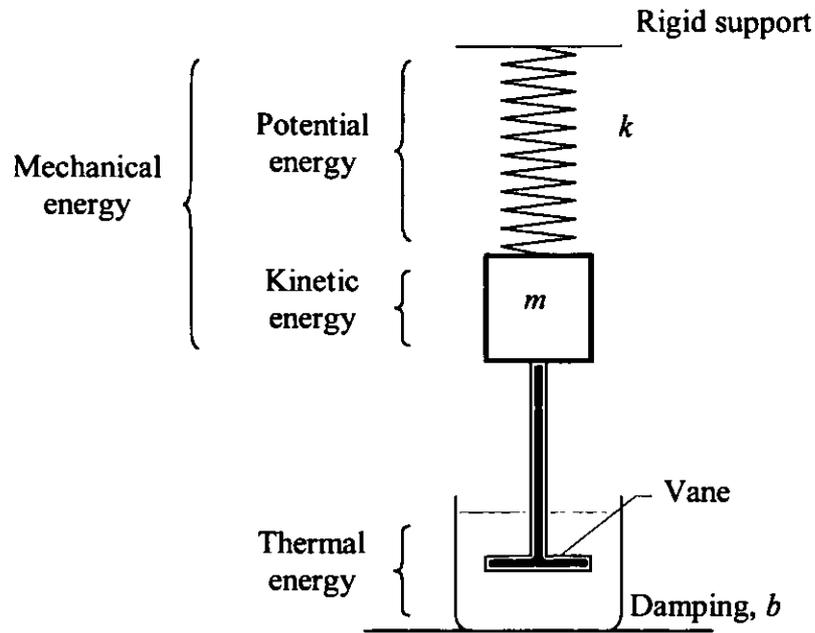


Figure 4-11 An idealised damped simple harmonic oscillator. A vane immersed in a liquid exerts, on the oscillating block, a damping force given by $-b(dx/dt)$, where b is the damping constant.

Let us assume that the liquid exerts a damping force F_d that is proportional in magnitude to the velocity v of the vane and block (an assumption which is accurate if the vane moves slowly). Then

$$F_d = -bv \quad (48)$$

where b is a damping constant that depends on the characteristics of the vane and the liquid, and the minus sign signifies that the force F_d opposes the motion. The resultant force acting on the block is then

$$\sum F = -kx - bv \quad (49)$$

or, since $v = \frac{dx}{dt}$,

$$\sum F = -kx - b \frac{dx}{dt} \quad (50)$$

Using Newton's second law (15),

$$m \frac{d^2 x}{dt^2} + b \frac{dx}{dt} + kx = 0 \quad (51)$$

it can be shown that the displacement as a function of time is

$$x(t) = x_m e^{-bt/2m} \cos(\omega_d t + \phi) \quad (52)$$

This solution is valid for $(b/2m)^2 < k/m$, where ω_d , the angular frequency of the damped oscillator, is given by

$$\omega_d = \sqrt{\frac{k}{m} - \frac{b^2}{4m^2}} \quad (53)$$

Equation (52) can be regarded as a cosine function whose amplitude is $x_m e^{-bt/2m}$, and decreases exponentially with time. The rate at which the amplitude decreases depends on the damping constant b , but not on the spring constant k (this can be seen in Figure 4-12). If $b=0$ (there is no damping), then (53) reduces to (44) for the angular frequency of an undamped oscillator, and (52) reduces to (43) for the displacement of an undamped oscillator.

As the damping is increased in the system, the oscillatory motion will become more and more damped, until, eventually, a certain value of the damping will be reached at which the mass will no longer oscillate when released following a disturbance. Instead it will return very slowly towards the initial equilibrium position, the rate of approach decreasing the nearer it gets. This is a non-oscillatory or aperiodic motion Figure 4-13a and the smallest value of the damping that will produce this result is called the *critical damping* b_{crit} , given by:

$$b_{\text{crit}} = 2\sqrt{mk} \quad (54)$$

In Figure 4-12 (bottom left plot, $m=1$, $k=1$ and $b=2$), it can be seen that since the damping equals the critical damping ($b_{\text{crit}}=2$) the motion is non-oscillatory. Values of b smaller than b_{crit} produce *underdamped motion*, that is, oscillation with exponentially decreasing amplitude.

The solution of (50) is qualitatively different for a large damping. If $(b/2m)^2 > k/m$ (i.e. values of b greater than b_{crit}) then the damping force effectively prevents oscillations, and the motion is called *overdamped*. As shown in Figure 4-13b, an overdamped oscillator initially displaced from equilibrium slowly approaches the equilibrium position without passing through it, i.e., continuously decreasing amplitude without oscillation. Any further increase of damping results in a slower rate of decay of the aperiodic motion.

A critically damped oscillator does not oscillate, but approaches the equilibrium position more rapidly than the overdamped and the underdamped oscillator.

For an undamped oscillator, the mechanical energy is constant and is given by

$$E = \frac{1}{2}kx_m^2 \quad (55)$$

If the oscillator is damped, it is a non-conservative system, and the mechanical energy is not constant but decreases with time. If the damping is small ($b \ll \sqrt{km}$), then $\omega_d \approx \omega$, where ω is the angular frequency of the undamped oscillator. For small values of b , the mechanical energy E of the oscillator is given by

$$E(t) = \frac{1}{2}kx_m^2 e^{-bt/m} \quad (56)$$

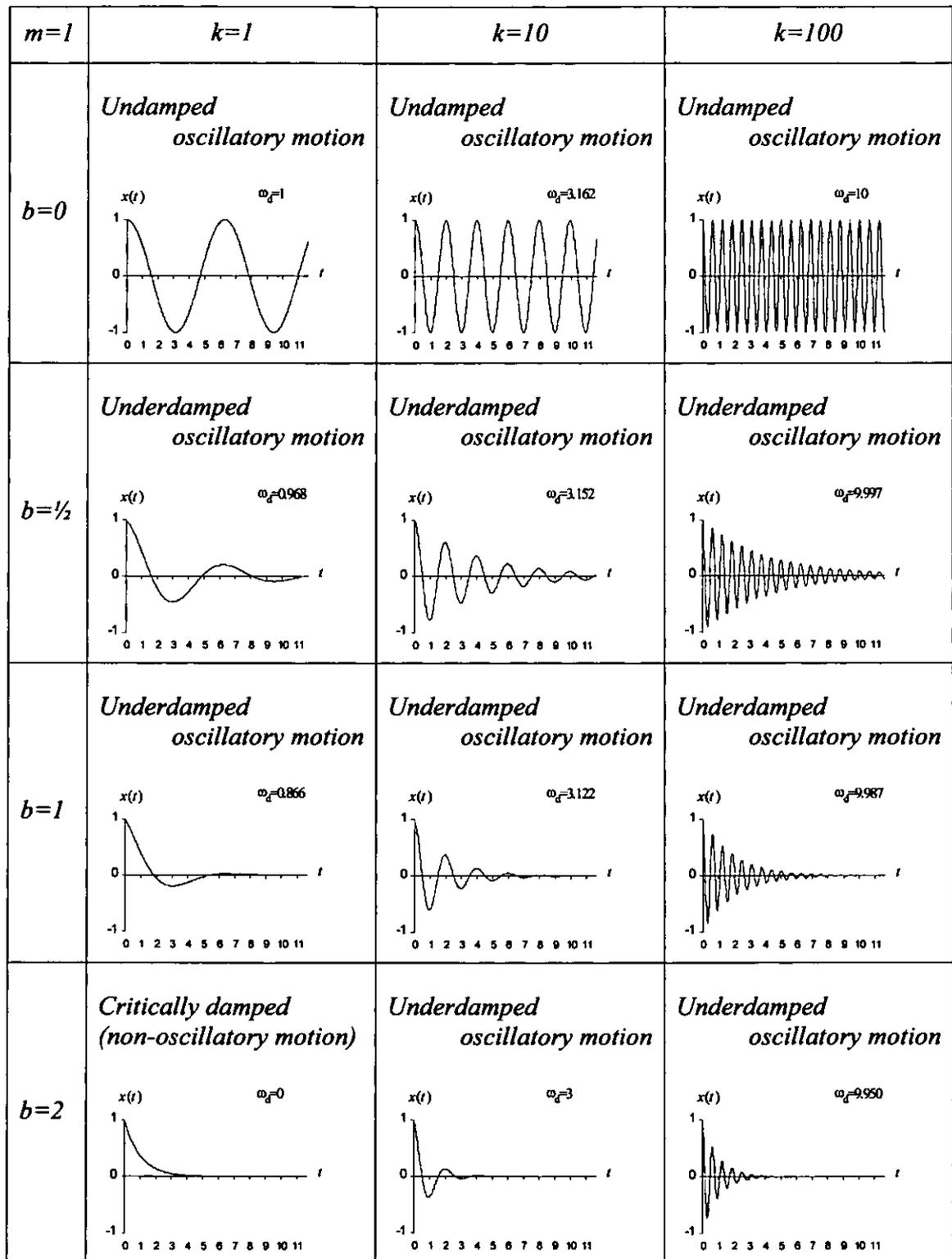


Figure 4-12 Motion of the damped oscillator for different values of damping constant b and spring constant k . SI units are assumed.

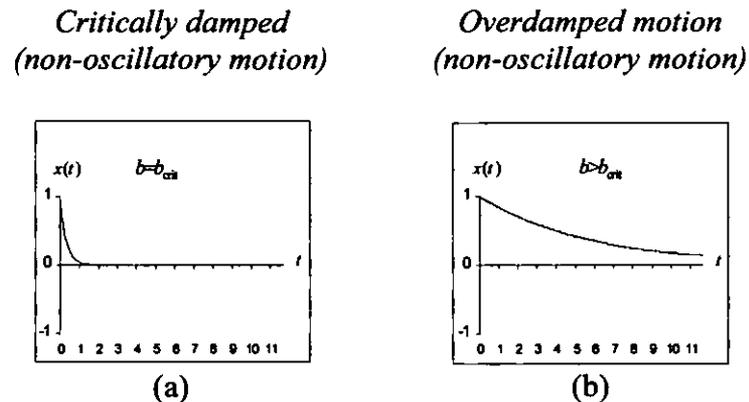


Figure 4-13 (a) Fastest convergence is given for critically damped motion $b=b_{\text{crit}}$, and (b) overdamped motion $b>b_{\text{crit}}$.

From the analysis of the damped harmonic motion it can be concluded that:

1. If the system is damped, the initial mechanical energy is eventually dissipated in the form of thermal energy, leaving the mass at rest.
2. According to equation (56), the time for the system to reach the equilibrium position depends on the damping constant b . The fastest convergence is given for the critical damping $b_{\text{crit}} = 2\sqrt{mk}$.
3. The final position of the mass, that is, when the equilibrium is reached, does not depend on the values of the mass m nor the spring constant k nor the damping constant b .

4.4.5.3 Image System

In the image system, all pairs of neighbour images are hinged together forming a network of connected images. The topology of the network depends on the path followed by the camera when acquiring the image sequence. It could be as simple as a straight path where each image is connected only to the previous and to the next image in the sequence (excluding the first and the last images), or as complex as an erratic sequence around a point in the scene in which case there will be a high concentration of images overlapping extensively, and each of these images will have several neighbours.

Two images linked together using the hinge constraint behave as two masses connected by two springs as shown in Figure 4-9. Although the problem is more complex than in the damped spring-mass system since the motion of the images is three-dimensional and the images are connected by two springs, the type of motion will fall into one of the reviewed categories. That is, the motion of the images will be either overdamped, critically damped or underdamped, being of an oscillatory nature in the latter case. Critically damped motion would be desirable, since it provides the fastest convergence to the equilibrium state, therefore, appropriate values must be chosen for the spring constant and the damping constant.

The role of the springs is to attract the images to their relative correct positions and since no information on the accuracy of each individual alignment is used at present, all springs will be given the same spring constant.

Since we are not interested in the trajectory of the images but in their final equilibrium positions, the value of the spring constant k is not significant as long as equilibrium is eventually reached.

The damping agent necessary to prevent unlimited oscillations can be thought of as a viscous fluid in which the network of connected images is submerged, the damping forces exerted by the fluid being proportional to the velocity of the bodies moving in it. The angular velocity of the images must be damped as well, hence a suitable drag torque proportional to the angular velocity will be present in the model. This drag prevents oscillatory rotations of the images.

It should be noted that friction only exists between the images and the fluid, there is no friction between the images. Images can share the same region of the space with other images and they can pass unhindered through each other. Springs are massless and dimensionless more like the invisible gravitational forces. This is important because when picturing the image system in one's mind it may seem that the images could end up entangled in a mess of real springs and solid rectangular planes, and this is not the case.

The mechanical energy of the system is the sum of potential and kinetic energies. Kinetic energy exists in the moving images, and the potential energy is stored in the springs. When the images are initially positioned at rest, the mechanical energy equals the potential energy since the kinetic energy is zero. The images will then start to move towards a lower potential energy position and the system will dissipate the excess energy and reach an equilibrium position where the net force on each image is zero.

The kinetic energy at the equilibrium position is zero since the images are not moving, but the potential energy is not zero (unless all the alignments were consistent). An internal tension between images is present at any of the equilibrium positions. The oscillation of potential and kinetic energy is an indication of oscillation in a system with several masses.

The number of different possible equilibrium positions depends on the configuration of the network and the initial position of the images. To ensure that the simulation does not stop at a local minimum, auxiliary constraints other than the hinge constraint need to be introduced into the model which prevent the occurrence of such cases. These constraints are explained in sections 6.4.2.1 and 6.4.2.2.

The error present in a pair of neighbour images i is given by (41), it represents the quality of the positions of the images in pair i with respect to its computed registration. The maximum error, which is the error for the worst case of image pairs, is given by

$$E_{\max} = \max \{E_1, E_2, \dots, E_M\} \quad (57)$$

where M is the number of neighbour pairs of images, and the average error is given by

$$E_{\text{aver}} = \frac{\sum_{i=1}^M E_i}{M} \quad (58)$$

Initially, the images are positioned sequentially in a plane. The system has a potential energy stored in the elongated springs and no kinetic energy since it is at rest. When the simulation is run, the images start moving, the kinetic energy increases and the potential

energy decreases accordingly. Some of the energy is dissipated due to the damping which subtracts energy from the moving images.

The simulation can be run until the kinetic energy is totally dissipated. There will remain some potential energy in the system unless all the alignments were error-free. The quality of the resulting distribution of the images can be assessed by looking at the maximum error (57) and the average error (58).

4.4.5.4 Example

The following figures show the progression of a pair of overlapping images initially placed at a distance from each other. In Figure 4-14 the images appear in the initial position. The forces exerted on the hinge points pull the images together.

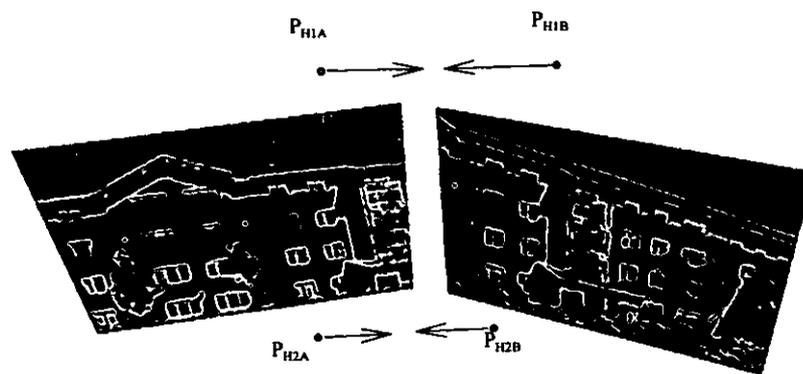


Figure 4-14 Initial position. The arrows represent the forces pulling the hinge points together.

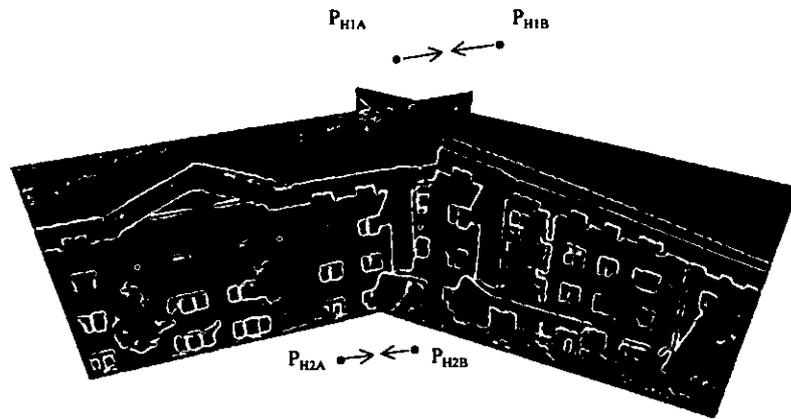


Figure 4-15 After a few iterations the hinge points are closer to each other. Since the distance between hinge points is smaller than in Figure 4-14, the forces pulling the hinge points together are themselves smaller.

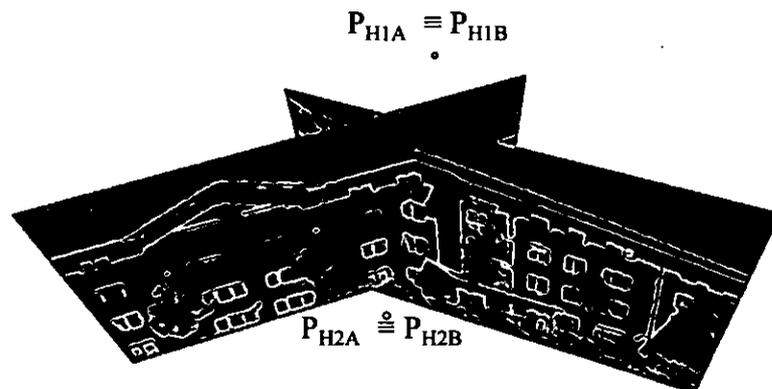


Figure 4-16 At the equilibrium position, the hinge points overlap perfectly. In this position, the pixel intensities at the intersection are common to both images.



Figure 4-17 The aligned images as seen from a viewpoint directly in front of both images.

Figure 4-16 shows one particular solution that satisfies the hinge constraint. However, unless additional information is determined regarding the angle between the two images, it is not possible to fully establish the position and orientation of the images. It is therefore necessary to introduce additional constraints into the model. For instance, the angle between image planes in a planar mosaic (where the scene is 2D), is known to be zero. In the cylindrical or the spherical mosaic, a spherical constraint is designed to guide the images into the designated shape.

4.5 Summary

The looping path problem has been defined and previous approaches to its solution have been reviewed. The author has presented an original solution which makes use of a physical simulation to consistently align the images in a mosaic. The concepts of physics which are needed to implement such a solution have been discussed and a physical model for the images has been presented which addresses the problems of stability and convergence. Experimental results are shown in the next two chapters.

Chapter 5 PLANAR SCENE MOSAIC

5.1 Introduction

A planar scene mosaic is assembled from a set of images recorded from a planar scene (e.g. a document, paint or blackboard) with a translating camera which keeps its sensor plane parallel to the scene at a fixed distance from the scene (as in Figure 3-3). The looping path problem can be present even in this simple type of mosaic, and since the planar mosaic involves only a limited number of parameters, it was initially chosen to investigate the adequacy of a physical analogy to solve the LPP.

In this chapter the practical implementation details are discussed. The experiments carried out involving planar scenes will be described and the resulting mosaics will be shown.

5.2 Image Acquisition

The experiments were carried out on document mosaicing, the document was a printed page. The images were acquired using a black and white video camera fixed over a board where the document was laid flat. This set-up is shown in Figure 5-1. The document was passed under the camera so that different portions of it fell within the field of view.

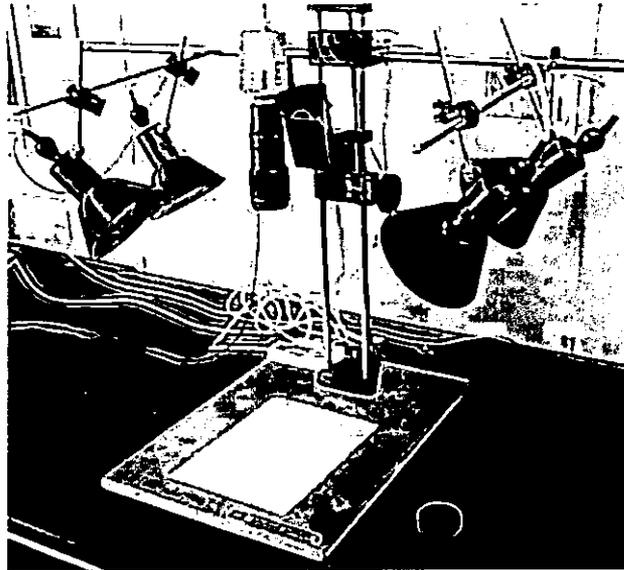


Figure 5-1 Camera set-up for document mosaicing.

With images obtained from a video camera (at a typical rate of 25 images per second), the maximum expected translation in pixels between successive images depends on the velocity of the camera. It also depends on the zoom settings, for with high magnifications small camera movements produce large motions in the images. Large inter-frame motions can produce motion blur (see section 3.2.4.4).

Another problem associated with video is that it is an interlaced scanning system, i.e. each image is composed of two fields which are captured sequentially in time (see section 2.2.6). Motion artifacts occur when the two fields are displayed together.

In order to use the full resolution of each frame, the images were recorded while the document was stationary. A series of snapshots were taken modifying slightly the position of the document in between. In effect, the video camera was used as a still digital camera.

Care must be taken to ensure that the images captured are sufficient to complete a mosaic of the whole document without leaving uncovered spaces. This was achieved by planning the acquisition in advance, testing for reasonable overlapping between adjacent path sections.

5.3 Implementation of Image Registration

This section discusses the registration technique chosen and its implementation. It finds the rigid transformation that aligns two images via direct estimation, reducing the computational cost by means of hierarchical pyramids, rotation by mapping Cartesian to polar coordinates, progressive complexity models, and achieving subpixel accuracy by extrapolation.

5.3.1 Transformation Model

The rigid model has been chosen to align two overlapping images. This three-parameter transformation model can describe translations and rotations in the scene plane, though it does not account for scale changes. The rigid model correctly describes the transformations between two images acquired as mentioned in the introduction to this chapter.

Since only three parameters are involved in the rigid model, there is less margin for inaccuracies. These parameters are horizontal translation Δx , vertical translation Δy and rotation angle $\Delta\theta$. The function f_p maps corresponding pixels in the images according to a set of parameters $P=\{\Delta x, \Delta y, \Delta\theta\}$.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = f_p(x, y) = \begin{pmatrix} \cos \Delta\theta & -\sin \Delta\theta & \Delta x \\ \sin \Delta\theta & \cos \Delta\theta & \Delta y \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (59)$$

In addition, the rigid model can be used as an approximation to perspective projection, and it is adequate when the effective focal length is sufficiently large (narrow field of view), and the camera is not panning excessively [Mann and Picard, 1994c] [Mostafavi and Smith, 1978]. The registration needs to be accurate only through the *intersection line* (see section 4.4.4) since misalignments will be visible only at the intersection line between neighbour images.

5.3.2 Feature Space

Rather than identifying salient features such as corners or lines, the image registration method that has been implemented makes use of pixel intensities directly. This method is referred to as *direct estimation*. For this feature space to provide suitable information, it only needs intensity variations in the images. Direct estimation was found to be an appropriate method to use because of its robustness over feature correspondence methods (particularly in the presence of lens distortion) and for not requiring the presence of any particular feature in the images.

The full overlapping area common to both images can be used in the registration, however lens aberrations such as distortion and coma (see section 2.2.8) are less acute in the centre of the image as compared to the pixels in the boundaries. According to [Mostafavi and Smith, 1978], in the presence of a geometric distortion, there is an image size that minimises the probability of false acquisition. This optimum image size decreases with increasing geometric distortion. In addition, using the central region of the each image is equivalent to reducing the field of view. This reduces the error due to approximating the projective transformation with the rigid transformation. Therefore, a square mask M centred in the overlapping area of the two images was used in the computation of the similarity measure. Only the overlapping pixels covered by the mask are used to compute the similarity measure. Thus the region over which a similarity measure is evaluated is defined by

$$S = M \cap A \cap B \quad (60)$$

The shape of the overlapping region between the images may be non-square for $\Delta\theta \neq 0$ or for large translations. Figure 5-2 is an example in which S is not square because it is clipped by the edge of *image B*.

The results using both similarity measures were compared in test images and were only different by a small fraction of a pixel, therefore the latter was adopted for being computationally less expensive.

In RGB encoded images, each pixel is represented by three values, one for each of the red, green and blue colour components. The similarity measure for a colour image was calculated as the average of individual similarity measures computed for each colour component.

$$D_{\text{colour}} = \frac{D_{\text{R}} + D_{\text{G}} + D_{\text{B}}}{3} \quad (63)$$

Since the number of overlapping pixels may vary for different parameters of the transformation \mathbf{P} , for the purpose of comparison, the similarity value D_{colour} has to be normalised by dividing it by the number of pixels in \mathbf{S}

$$D_{\text{norm}} = \frac{D_{\text{colour}}}{n_{\mathbf{p}}} \quad (64)$$

where $n_{\mathbf{p}}$ is the number of pixels in \mathbf{S} .

A typical example of the similarity map of two overlapping images for $\Delta\theta = 0$ is shown in Figure 5-3. The yellow spot indicates the values of Δx and Δy for which the similarity is maximum.

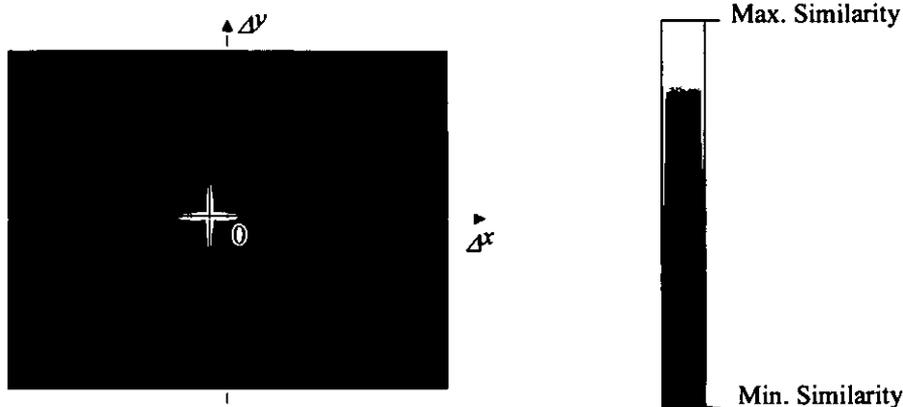


Figure 5-3 Similarity map of two overlapping images for $\Delta\theta = 0$.

5.3.4 Search Space

The search space is the set of all possible transformations, that is, all possible combinations of Δx , Δy and $\Delta\theta$. An estimation of the maximum expected translation and rotation between the images is necessary in order to define the search space. These limits are defined during the image acquisition stage by limiting the movement of the camera to guarantee the necessary overlapping.

Another limiting factor on the search space arises when the transformation being tested involves large translations. In these cases, the size of the overlapping area may not be large enough to provide reliable results, and the similarity measure cannot be computed.

5.3.5 Search Strategy

The following sections explain the algorithms used for the computation of the translation and rotation components of the transformation that best aligns two images.

Testing all transformations in the search space (even with only 3 parameters) is computationally very expensive. A more efficient method for determining the best transformation within the search space has been implemented by using progressive complexity transformation models together with hierarchical pyramids and rotation matching based on polar to Cartesian coordinate conversion.

5.3.5.1 Hierarchical Pyramids

Hierarchical matching is a technique used to achieve shorter computation times (see section 3.2.3.4 for an introduction). Smaller images are created from the original images by averaging blocks of pixels. In the original image whose size is (*Width*, *Height*), blocks of 4 pixels are averaged resulting in an image of order 1 whose size is ($Width/2$, $Height/2$). Order 2 images are created by averaging blocks of 16 pixels, and order 3 images by using blocks of 64 pixels. An image of order n has a size ($Width/2^n$, $Height/2^n$), and every pixel represents the average value of a block of 2^{2n} pixels in the original image.

A translation of 5 pixels in an order 3 image corresponds to a translation of $5 \times 2^3 \pm 2^3$ pixels in the original image. The term $\pm 2^3$ indicates that actual the position in the original image is not fully determined but falls within the interval $[-8, 8]$ around the position 5×2^3 . Very large displacements can be found with the highest order images, then lower order images are used to increase the accuracy of the displacement. A description of this method can be found in [Hansen et al., 1994]. The registration of two original images is started by registering the smallest image size (highest order). This registration is used as an initial position to register iteratively lower order images until the registration is calculated for order 0, which corresponds to the original images.

The benefits of hierarchical registration are clear when comparing its computational cost with that of directly registering two full size (order 0) images. In the first case, to search for the best translation over a space of 128×128 pixels, the similarity measure would have to be computed 16384 times. However, only 304 computations would be necessary using the hierarchical approach. Table 5-1 shows the computational cost using the hierarchical images starting at order 3.

Order	Search Space (pixels)	Number of Computations of Similarity Measure
3	$128/2^3 \times 128/2^3 = 16 \times 16$	256
2	$16/2^2 \times 16/2^2 = 4 \times 4$	16
1	$8/2^1 \times 8/2^1 = 4 \times 4$	16
0	$4/2^0 \times 4/2^0 = 4 \times 4$	16
		Total 304

Table 5-1 Computational cost of hierarchical registration for a search space of 128×128 pixels.

This means that the search space can be dramatically reduced by starting the search with high order images and using the results as initial positions for the computation of the transformation at a lower order until order 0 (the original scale image) is reached.

The process for computing the translation between two images using hierarchical images is shown in Figure 5-4. The function **CALC_TRANSLATION** takes five parameters: the two images to align, the search space in pixels, and the initial translation to search from (Δx and Δy). The returned value is the translation in pixels that aligns the images supplied at each call to the function. The index in `image_i []` and `image_j []` denotes the order of the image.

```

trans3 = CALC_TRANSLATION(image_i[3], image_j[3], 16, 0,0)

trans2 = CALC_TRANSLATION(image_i[2], image_j[2], 4,
                           trans3.x*2, trans3.y*2)

trans1 = CALC_TRANSLATION(image_i[1], image_j[1], 4,
                           trans2.x*2, trans2.y*2)

trans0 = CALC_TRANSLATION(image_i[0], image_j[0], 4,
                           trans1.x*2, trans1.y*2)

```

Figure 5-4 Process to compute the translation using hierarchical images.

Actual values used in the program are shown in Table 5-2. Note the large search space at order 3 (smallest images). The search range values are referred to pixels in the images of the corresponding order, that is, a search space of 25×25 pixels in order 3 image corresponds to a search space of 200×200 pixels ($200 = 25 \times 2^3$) in the original size images. After this first extensive search, the search range will be limited to 4 pixels around the minimum.

Order	Search Space (pixels)	Square Mask (pixels)	Angle range (degrees)
3	25×25	15×15	$[-5, 5]$
2	4×4	25×25	$[-3, 3]$
1	4×4	50×50	$[-3, 3]$
0	4×4	100×100	$[-3, 3]$

Table 5-2 Values used in the registration program for an image size of 320×240 pixels.

5.3.5.2 Rotation

Rotating an image is a time consuming operation, in particular when subpixel accuracy by means of interpolation is necessary. The process that has been used for the

computation of the angle of rotation between two images involves the mapping of a circle onto a rectangle as shown in Figure 5-5. The warping takes a radial line from the source image (Figure 5-5a) and maps it into a row (in Figure 5-5c), corresponding directly to a mapping from the polar coordinate system to the Cartesian coordinate system, i.e. $(r, \theta) \rightarrow (x, y)$.

A vertical translation in the warped image (Figure 5-5c) is equivalent to a rotation in the original image (Figure 5-5a). Thus the problem of maximising the similarity function for different angles is reduced to a vertical translation matching, which is computationally less expensive and uses the same algorithms developed for registering the translation.

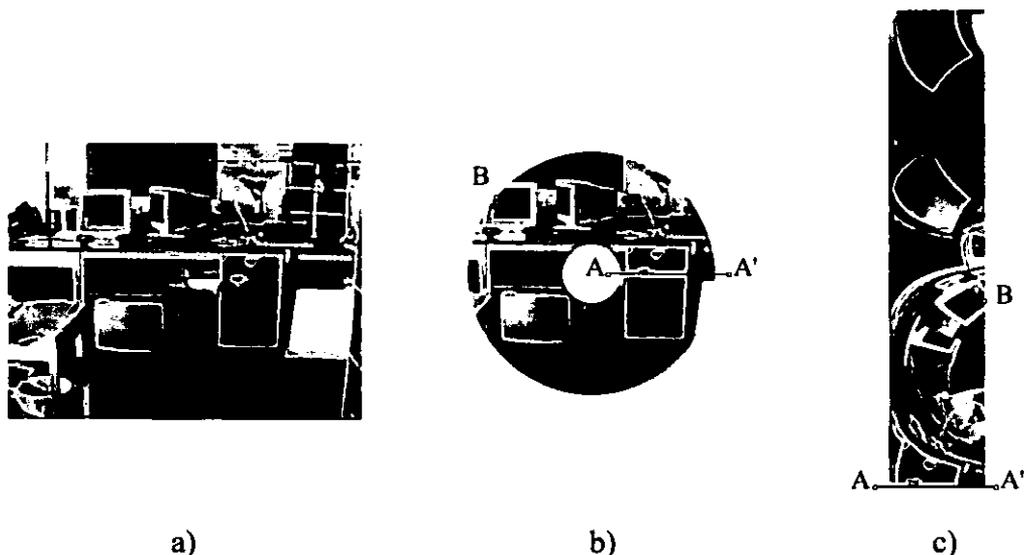


Figure 5-5 By warping from the polar to the Cartesian coordinate system, finding the alignment angle between two images is reduced to a vertical translation matching. (a) Original image. (b) Area to be warped onto a rectangle. (c) Effect of warping a ring onto a rectangle.

5.3.5.3 Image Registration with Progressive Complexity Models

It has been shown that in image alignment, the use of transformation models with progressive complexity reduces the computation cost [Sawhney and Kumar, 1997] [Hansen et al., 1994]. Although for the rigid transformation there are only three parameters to be computed, two for the translation and one for the angle of rotation, a

simpler model involving solely translation can be initially used for the alignment. Then, using this translation component, the angle of rotation can be computed. The process is summarised in Figure 5-6.

```

1.  DO {
2.      translation = REG_TRANSLATION(image i, image j)
3.      TRANSLATE_IMAGE(image j, translation)
4.      angle = REG_ROTATION(image i, image j)
5.      ROTATE_IMAGE(image j, angle)
6.  } UNTIL calculations arrive at the same results

```

Figure 5-6 Steps for iteratively calculating the best translation and rotation alignment parameters between two overlapping images using transformation models with progressive complexity.

This particular implementation starts by finding the translation (line 2 in Figure 5-6) that best aligns the pair of images. This offset however does not account for the rotation between the images, but for small angles it may be neglected at this stage. Then, using the current translation (line 3), the best rotation angle is searched (line 4). This process of adjusting translation and then rotation is repeated in several passes until better accuracy cannot be achieved. This typically requires only 2 or 3 iterations. The following example illustrates the registration process.

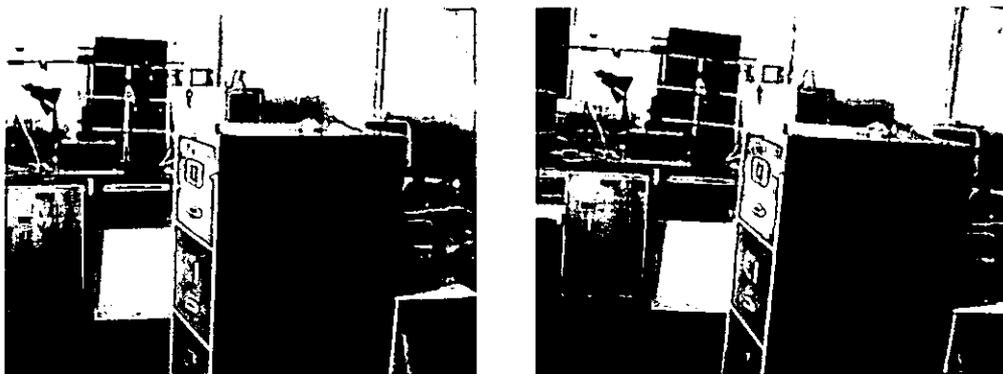


Figure 5-7 Test images.

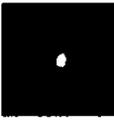
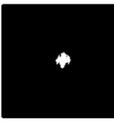
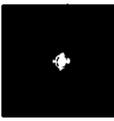
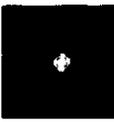
Iteration	Starting transformation	Similarity Measure Map			Result
		Order 3	Order 2	Order 1	
0	$\Delta x = 0$ $\Delta y = 0$ $\Delta \theta = 0$				$\Delta x = -28$ $\Delta y = 6$ $\Delta \theta = 0.027$
1	$\Delta x = -28$ $\Delta y = 6$ $\Delta \theta = 0.027$				$\Delta x = -27$ $\Delta y = 3$ $\Delta \theta = 0.047$
2	$\Delta x = -27$ $\Delta y = 3$ $\Delta \theta = 0.047$				$\Delta x = -26$ $\Delta y = 2$ $\Delta \theta = 0.058$
3	$\Delta x = -26$ $\Delta y = 2$ $\Delta \theta = 0.058$				$\Delta x = -26$ $\Delta y = 1$ $\Delta \theta = 0.066$
4	$\Delta x = -26$ $\Delta y = 1$ $\Delta \theta = 0.066$				$\Delta x = -26$ $\Delta y = 1$

Table 5-3 Registration process of the images in Figure 5-7.

The starting transformation for iteration 0 is $(\Delta x=0, \Delta y=0, \Delta \theta=0)$, and the resulting alignment is $(\Delta x=-28, \Delta y=6, \Delta \theta=0.027)$. This result is then used in iteration 1 as a starting transformation to obtain a more accurate alignment since the translation is now computed based on the images already rotated by 0.027 radians. Since the resulting

translation of iteration 4 is the same as for the previous iteration, the loop concludes with this final result.

In the previous example, there is only one minimum in the similarity measure map. However, this is not always the case. When a repetitive pattern is present in the images, there will be local minima for which this pattern is aligned.

To avoid arriving at an erroneous local minimum it was necessary to test all local minima at every pyramid level, and choose the global minimum at order 0. This problem was unveiled when registering document images, where false matches at the lower resolution levels may occur due to the repetitive nature of text lines. Figure 5-8 shows two overlapping images from a document and the similarity map. The number of local minima can be very high in this type of image, and the computation cost increases accordingly.

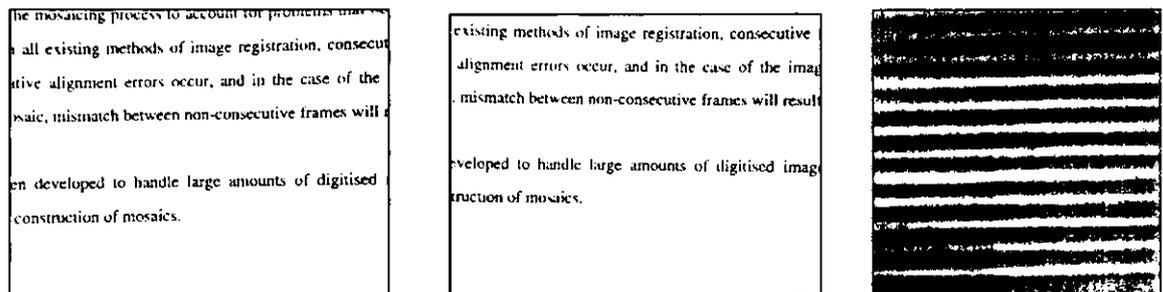


Figure 5-8 Overlapping document images and their similarity map for $\Delta\theta=0$.

Typically, the time required in the computation of the alignment of two overlapping images (320x240) in a Pentium 166 MHz is in the order of one to two minutes.

5.3.5.4 Subpixel Accuracy

The similarity map is computed for integer pixel translations, i.e. it is accurate to one pixel. Subpixel accuracy can be achieved by separately estimating the horizontal and vertical location of the minimum in the similarity measure map. Using the symmetric triangle model shown in Figure 5-9 [Nishihara, 1984], the equation for computing the position x of the estimated minimum error E_x is

$$x = i + \frac{E_{i+1} - E_{i-1}}{2(E_i - \min(E_{i+1}, E_{i-1}))} \quad (65)$$

where i is the integer translation for the minimum error E_i .

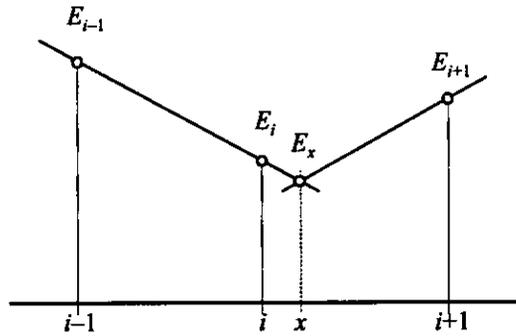


Figure 5-9 Subpixel accuracy.

Although the accuracy that can be achieved in image alignment is excellent when assessed by the human eye, it is not error-free. Due to quantisation and distortions, a small error is always present and, when accumulated, it will manifest itself in a looping path.

5.4 Physical Model for the Planar Mosaic

For a planar scene and a camera motion such that the plane of the image sensor remains parallel to the scene (as in Figure 3-3), the affine model correctly describes the transformation between two overlapping images.

Furthermore, the angle between the image planes is known to be zero, since all images lie on the same plane. Hence, the proposed method involving a physical simulation of the hinged image system can be used to ensure the consistency of the position of images on the mosaic.

In two dimensions the problem is simpler, the hinge model can be used since all forces acting in the model are coplanar ensuring that the images will never leave the initial plane.

The hinge model was not fully developed until it was necessary for the three dimensional case of images in different planes. Instead, at the time of implementing the physical model for planar mosaics a simpler model was used, obtained by joining the centres of the images with springs, and applying torques to each image according to its angular orientation with respect to its neighbours.

5.5 Image Integration

After the simulation was run and the positions of the images were established, the mosaic was ready to be rendered. The contribution of each image to the mosaic was determined by the Voronoi diagram. The Voronoi polygon associated to a particular individual image defines the area in the mosaic whose points are closer to this image than to any other individual image. Therefore, the Voronoi polygons are an efficient way of determining the contribution of each individual image to the mosaic by ensuring that the centres of the images are used rather than pixels distant from the centre. This benefits from the fact that in general, deficiencies in image quality caused by lens distortion and coma are worse in regions further from the centre of the image.

5.6 Experimental Results on Document Mosaic

The set-up shown in Figure 5-1 was used for acquisition of the images in this experiment. The document was passed under the camera and a total of 141 images were recorded. The path followed by the camera's view over the document is represented in Figure 5-10. Note that the looping path does not necessarily have to cross itself, but it is present whenever non-consecutive images (those which are distant in the sequence) do overlap.



Figure 5-10 Path followed by the camera over the document.

Figure 5-11 shows the mosaic obtained with successive image alignment only. The accumulation of the error is visible between the first image (top left) and the image at its right (top centre). It is also very noticeable at the end of the path (bottom left) where the error has been accumulated over a larger number of images.

This misalignment, however wrong, is not too severe, being only a few pixels after more than 80 images (from bottom-left to end of sequence). This proves the accuracy attainable by the implemented registration algorithm, estimated as less than 1/10 of a pixel. However, it should be noted that the random distribution of the error in the individual registrations can cancel itself to some extent as it is being accumulated.

The initial position of the images in the mosaic, i.e. using successive image registrations only, proves to be sufficiently accurate as to provide neighbourhood information prior to computation of the neighbour image registrations.

The resulting mosaic after consistent image alignment is shown in Figure 5-12. Attention should be drawn to the seams that were noticeable in Figure 5-11. A major improvement in quality has been achieved.

Also, the slight modification of the position and orientation of all the images, necessary to satisfy the non-consecutive image alignments, has not introduced any visible

deterioration at the seams between successive images. The quality of the alignments between successive images remains undiminished. This proves that error distribution is a practical solution to the looping path problem.

It is not possible to assess the quality of the alignments between pairs of images whose seam coincides with interline blank space. Therefore, the experiment was repeated using the same text document with an overlaid grid. The grid provides references to assess the quality of the alignment over all the seams, and not just those where a line of text was intersected. The mosaic obtained after successive image alignment only is shown in Figure 5-13, and the mosaic after the consistent image alignment in Figure 5-14.

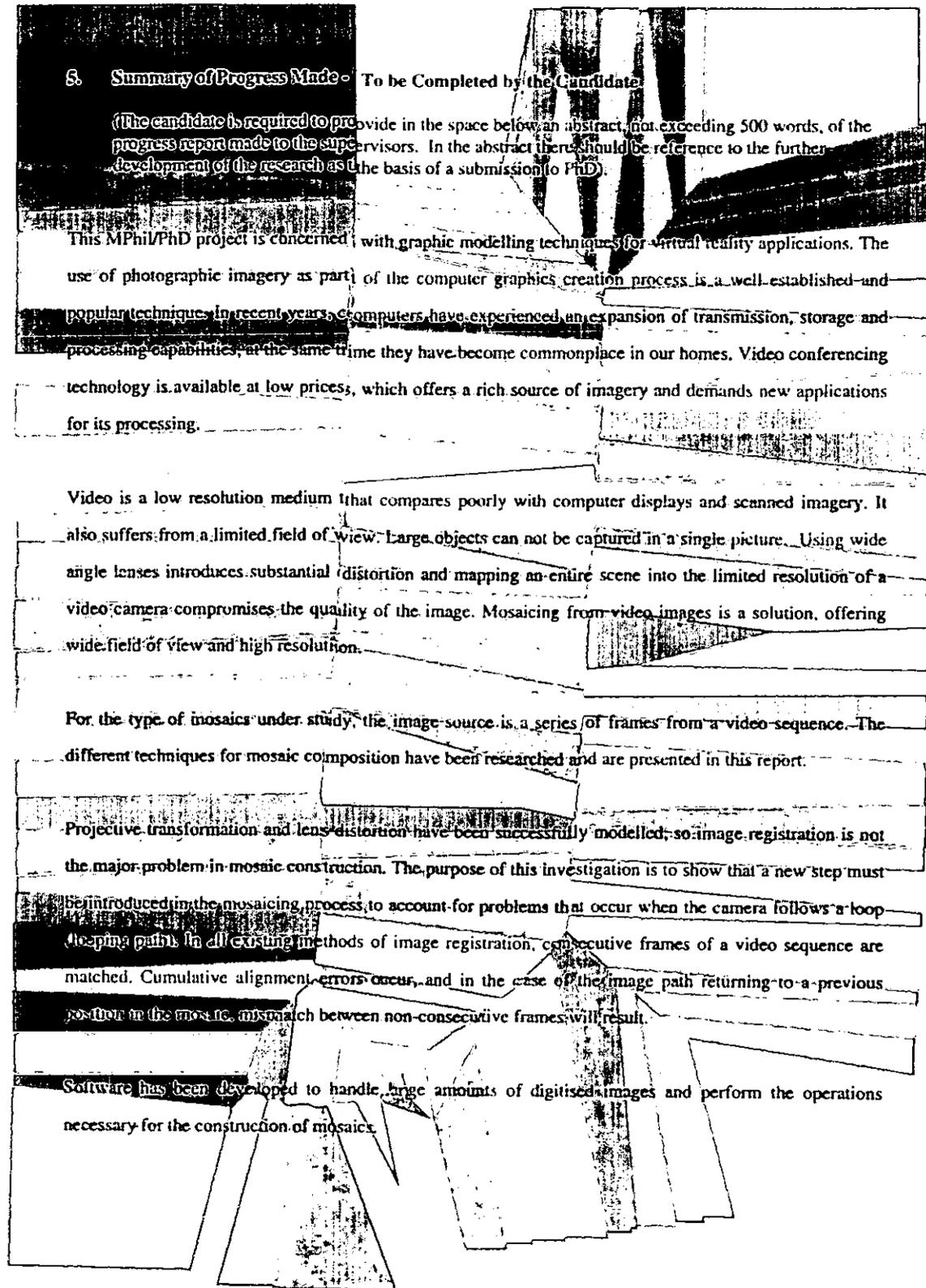


Figure 5-11 Mosaic with successive image alignment only. Note misalignments.

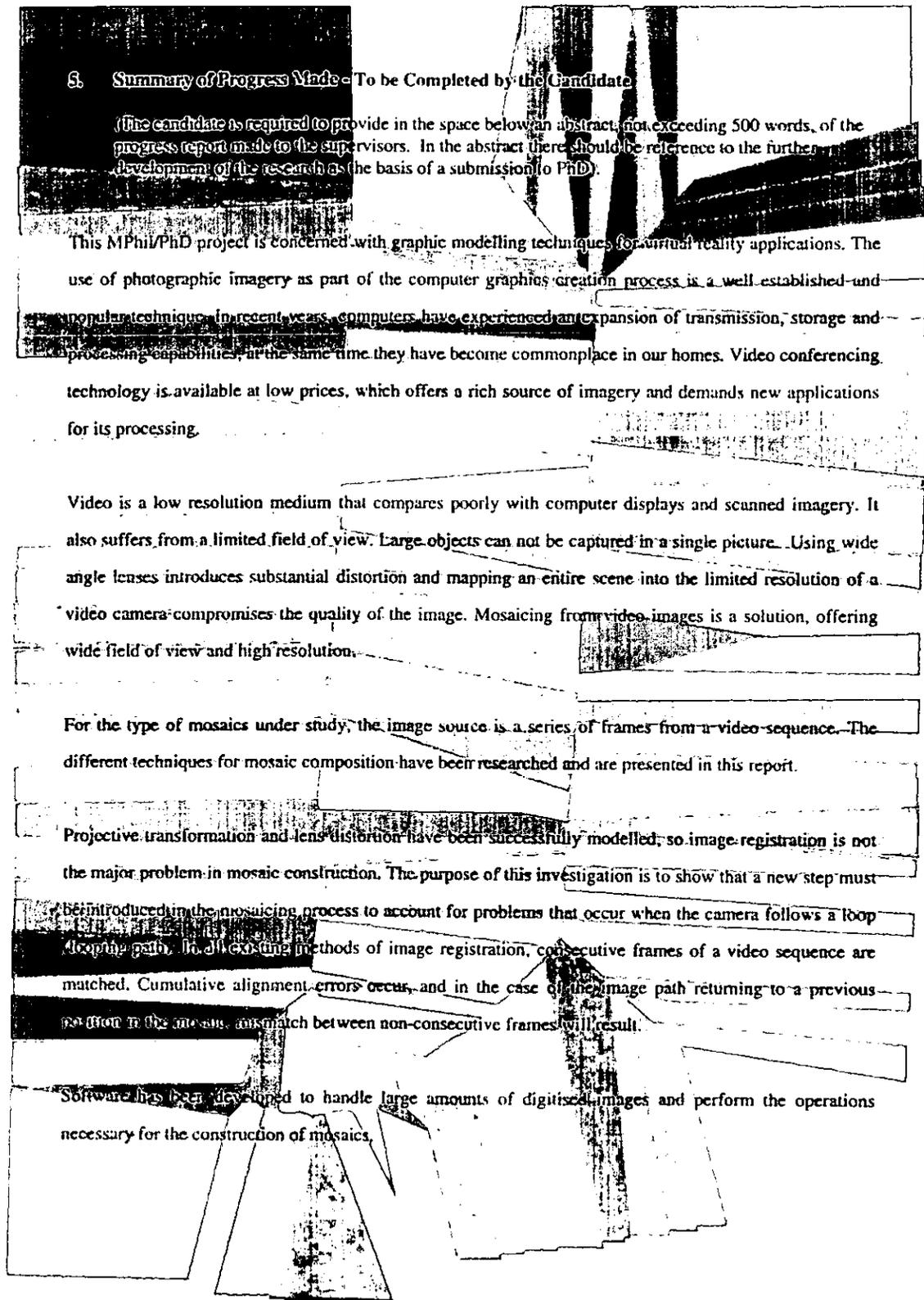


Figure 5-12 Mosaic after consistent image alignment. Compare to Figure 5-11.

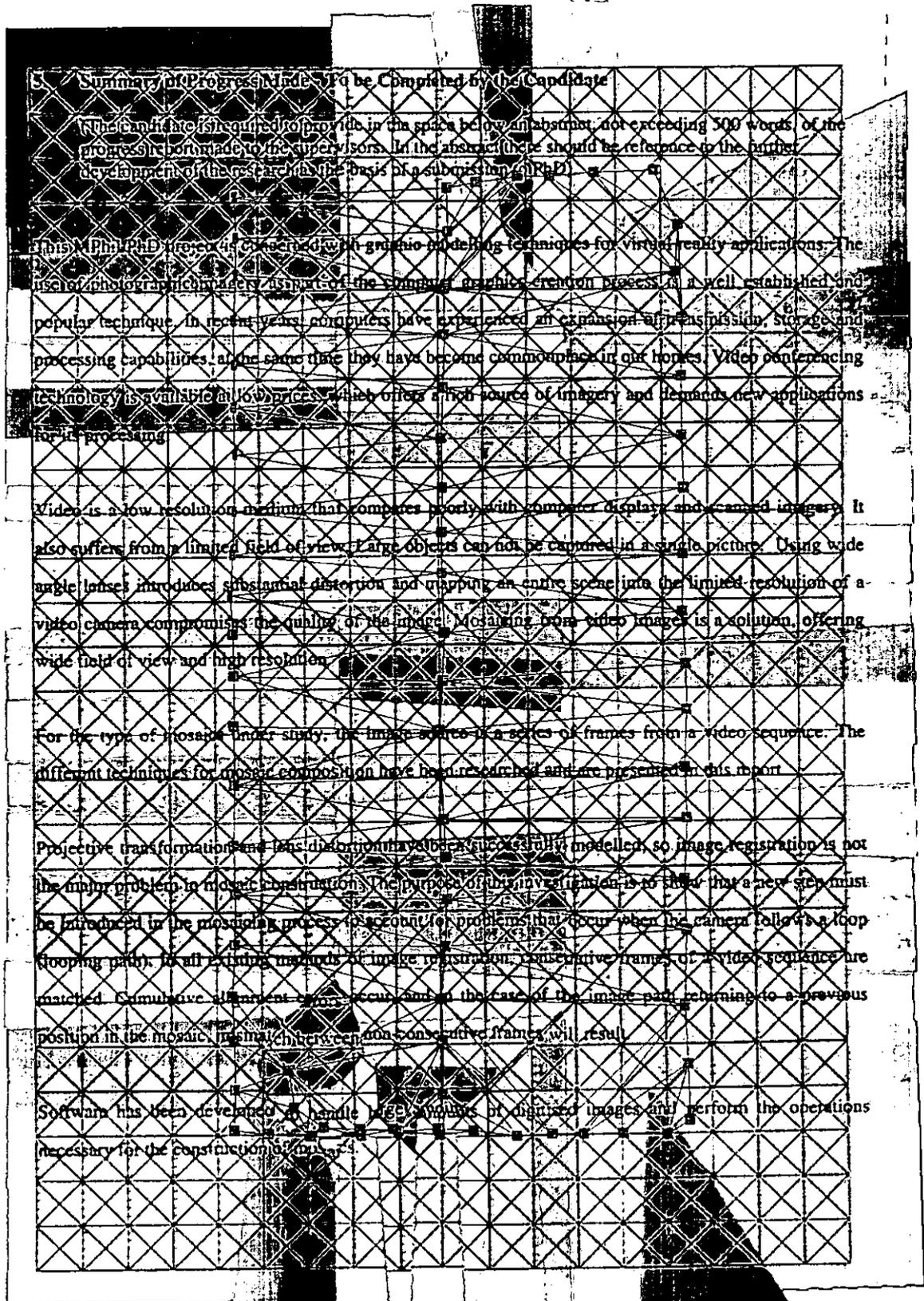


Figure 5-13 Mosaic with successive image alignment only. Note misalignments. The grid provides a means for assessing the quality of the alignments.

5.	Summary of Progress Made - To be Completed by the Candidate
(The candidate is required to provide in the space below an abstract, not exceeding 500 words, of the progress report made to the supervisors. In the abstract there should be reference to the further development of the research as the basis of a submission to PhD).	
<p>This MPhil/PhD project is concerned with graphic modelling techniques for virtual reality applications. The use of photographic imagers as part of the computer graphics creation process is a well established and popular technique. In recent years, computers have experienced an expansion of transmission, storage and processing capabilities, at the same time they have become commonplace in our homes. Video conferencing technology is available at low prices, which offers a rich source of imagery and demands new applications for its processing.</p>	
<p>Video is a low resolution medium that compares poorly with computer displays and scanned imagery. It also suffers from a limited field of view. Large objects cannot be captured in a single picture. Using wide angle lenses introduces substantial distortions and mapping an entire scene into the limited resolution of a video camera compromises the quality of the image. Mosaicing from video images is a solution, offering wide field of view and high resolution.</p>	
<p>For the type of mosaic under study, the image source is a series of frames from a video sequence. The different techniques for mosaic composition have been researched and are presented in this report.</p>	
<p>Projective transformation and lens distortion have been successfully modelled, so image registration is not the major problem to mosaic construction. The purpose of this investigation is to show that a new step must be introduced in the mosaicing process to account for problems that occur when the camera follows a loop (looping path). In all existing methods of image registration, consecutive frames of a video sequence are matched. Cumulative alignment errors occur and in the case of the image path returning to a previous position in the mosaic, mismatch between non-consecutive frames will result.</p>	
<p>Software has been developed to handle large amounts of digitised images and perform the operations necessary for the construction of mosaics.</p>	

Figure 5-14 Mosaic after consistent image alignment. Compare to Figure 5-13.

Mosaics offer essentially unlimited resolution. In this particular experiment, an image resolution of 1481×2139 pixels was obtained from smaller 736×560 pixel images. Any path can be followed for the capture of the individual images, and, by using a more powerful zoom lens or by placing the camera closer to the document, greater detail can be achieved.

Scanning can be performed at virtually any resolution. However, image registration cannot be performed without sufficient texture in the images. For example, if the camera zooms in into a white space in the document, it is not possible to compute its alignment with respect to another white image. Means of determining the validity of the registrations are necessary if the process is to be automated. The registrations were verified and corrected manually where the algorithm failed (mainly due to lack of features).

5.7 Summary

This chapter has described the implementation details of image registration and also shown an example of document mosaicing where the looping path caused visible distortion. Since the looping path problem has been solved, it is now possible to construct large mosaics composed of many images recorded in any order.

The proposed method produces its most interesting results in the construction of spherical mosaics, in which the images are distributed in three-dimensional space on the surface of a sphere. This is described in the next chapter.

Chapter 6 SPHERICAL MOSAIC

6.1 Introduction

In this chapter, three experiments are described in detail. These are progressively complex, and provide a structure upon which the relevant topics can be discussed conveniently. The final experiment represents the fulfilment of the full spherical mosaic and serves as proof of the usefulness of the approach presented in this thesis.

6.2 Image Registration Considerations

The image registration implementation explained in detail in section 5.3 has been used in the following experiments.

Images taken from the same viewpoint by panning or tilting a camera are related by 2D projective transformations. Thus, the image registration implementation, explained in detail in section 5.3, has been used in the following experiments. The rigid model will be used as an approximation to perspective projection. This approximation is valid for a narrow field of view and small displacements between the images (see section 5.3.1). Consecutive video images overlap extensively having small displacements, however, distant neighbour images present larger displacements. Large displacements can invalidate the approximation leading to errors in the registration. More accurate registration should be achievable with a technique optimised for finding the *intersection line* defined for the hinge constraint. Szeliski and Heung-Yeung [Szeliski and Heung-Yeung, 1997] have proposed a registration method which directly recovers rotations (pan and tilt angles) instead of general 8 parameter perspective transforms. These rotations effectively determine the intersection line.

The results obtained by automatic registration with the implemented technique were validated manually, and in the cases where they were wrong, the values were corrected manually. The main reasons for mismatch were: 1) very large displacements, i.e. small overlapping areas, and 2) lack of significant reference points on the images making it difficult to register even manually.

6.3 Partial Spherical Mosaic Mapped to a Plane

This section describes the creation of the first mosaic in this project. Although the mapping of a three-dimensional scene to a plane does not give good results, it has been included in this chapter since it shows the progression in the research towards more accurate models. The need for a three-dimensional spherical distribution of the images was seen during the execution of this experiment.

6.3.1 Image Acquisition

In this experiment, a handheld camcorder was connected to a frame grabber in a personal computer in which the images were stored at a low frame rate. The translation of the camera was kept to a minimum while it was panned over 90° and tilted about 45°. A total of 130 images were recorded.

A number of problems were encountered during image acquisition. These include inter-frame motion, parallax, and changes in image intensity due to the automatic gain control of the camera.

Video frames consist of two interlaced fields, and for a moving camera, inter-field motion becomes a problem (see section 2.2.6). The solution adopted was to use only one field per frame, discarding the other. The images were resampled horizontally to preserve the original aspect ratio. Thus, the useful resolution of each image was reduced to 368×288 pixels.

A handheld camera undergoes involuntary translations that introduce motion parallax when the scene contains objects near the camera (see section 3.2.4.1). In theory, panoramas can only be constructed if all images are taken by a camera whose optical

centres never moves. In practise, this depends on the amount of camera translation relative to the nearest object in front of the camera [Szeliski and Heung-Yeung, 1997].

Additional problems were created by the Automatic Gain Control (AGC) of the video camera since the camera determines the dynamic range needed for each frame by integrating the light over the image and not over the whole scope of the mosaic. Whenever possible, it is advisable to turn this function off and select the gain manually to satisfy the entire scene.

6.3.2 Physical Model for Image Alignment

The physical model used for this experiment is the same one used for the planar mosaic. That is, the mosaic will be composited on a plane (i.e. angle between image planes = 0°). Clearly, the camera motion is not appropriate for a planar scene. If the mosaic is composited on a plane, there is a projective distortion of the same type as the distortion found in world maps, where a sphere is projected on a plane.

In order to understand the implications of constructing the mosaic on a plane rather than on a sphere, the reader is challenged to answer the following. A bear sets off from home and travels 10 miles South, then walks 10 miles East, then turns North and runs for another 10 miles. The bear stops and realises that it is back home! Question: What colour is the bear?

Obviously, the answer is that the bear is a polar bear and its colour is white. Why? The North Pole is the only possible location on earth at which the bear could have started its journey without ending up about 10 miles East from where it started. Similarly, when a camera initially pointing up (optical axis vertical) tilts down to the horizontal level, then pans right and then tilts up (see Figure 6-1), the composition of images on a plane will incorrectly form a 'U' shape. If however, they are projected on a sphere, the first and final image would overlap at the tip of a curved triangle.

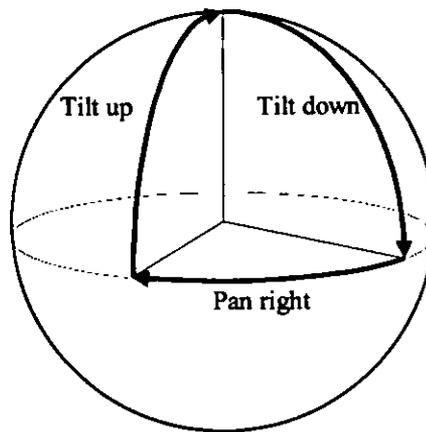


Figure 6-1 Path followed by a rotating camera represented on the surface of the sphere.

6.3.3 Image Integration

The integration of the images in the mosaic follows the same steps described for the planar scene (section 5.5), where the contribution of each image to the mosaic is determined by the Voronoi diagram, which selects the most central parts of the images for integration in the mosaic. Besides reducing distortions due to the lens, the seams between neighbour images coincide with the hinge intersection line, thus the joint between images occurs at the best position.

6.3.4 Results

The resulting mosaic after successive image alignment only can be seen in the introduction to Chapter 4, Figure 4-1 repeated as Figure 6-2, where it is used as a good example of the looping path problem. The path followed by the camera is represented with a red line starting from the left, and the misalignment is obvious at the intersection and at the end of the path, where the error has been accumulated over a larger number of images.

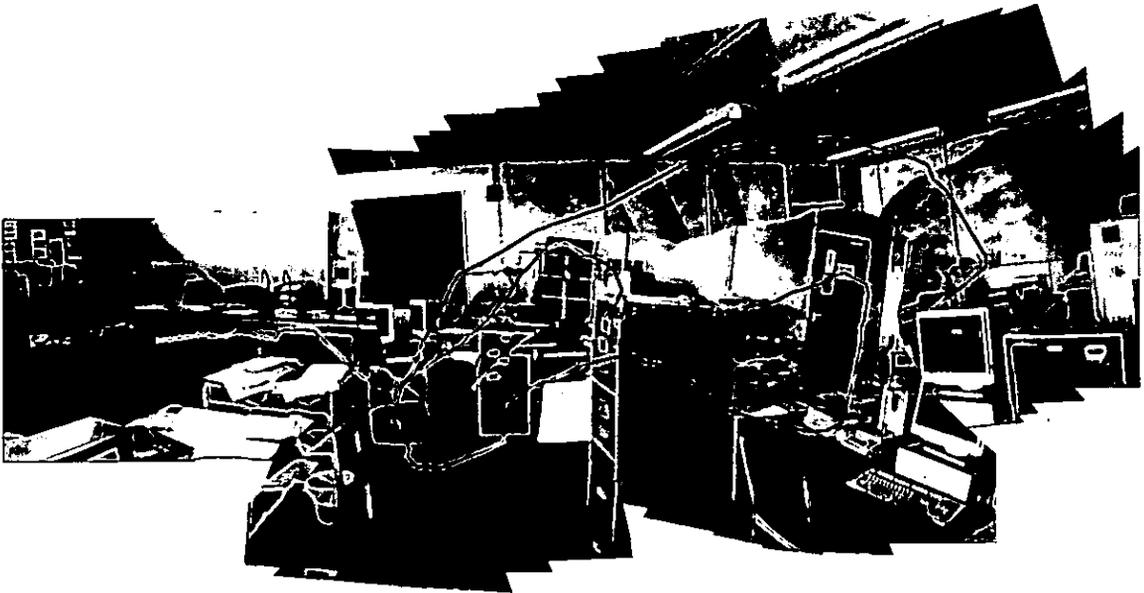


Figure 6-2 Mosaic resulting from successive image alignment only.

The severe misalignment found in this mosaic can be explained partially because of the misregistration caused by parallax, particularly between images containing the grey filing cabinet in the centre of the room. The main reason for this inconsistency is the mapping of a spherical distribution of images onto a plane without correction, i.e., constructing the mosaic on a plane rather than on a sphere.

Despite the severe misalignment, neighbour information is sufficient to update correctly the list with the new neighbours and compute their alignments.

Figure 6-3 shows the resulting mosaic after consistent image alignment. The improvement with respect to Figure 6-2 is substantial. However, certain types of distortion are present.

Differences in image brightness due to the camera's AGC have not been corrected. The effects of this can be seen where sudden changes in image brightness reveal the seams between images.

The largest distortions are present at the top of the grey filing cabinet, where individual misregistrations were caused by parallax.

Mapping the spherical distribution of images onto a plane without correction is the cause of small distortions. This is mainly shown between images located at the ceiling of the room.



Figure 6-3 Result of consistent image alignment applied to mosaic construction.

6.3.5 Summary

The execution of this experiment revealed a number of image acquisition problems. It has also shown that the construction on the plane of a mosaic without rectification is somehow valid for a small field of view.

The improvements achieved with respect to successive image alignment only, proves the applicability of error distribution to the problem of consistent image alignment.

6.4 Double Band Panoramic Mosaic

An enhanced panoramic mosaic, created by panning a camera 360° twice at two different tilt angles (i.e. two bands), is presented in this experiment.

While a single-band panoramic mosaic can be created by aligning the last and the first images, a two-band panorama like the one presented in this experiment, needs a more elaborate technique to ensure consistency between the two bands.

The physical model described in this experiment contains all the necessary elements to create full spherical mosaics. However, with the cylindrical shape of a panoramic mosaic, the behaviour of the model and the new constraints can be logically introduced into the discussion.

6.4.1 Image Acquisition

For this experiment, a handheld semi-professional video camera was used which recorded on SVHS tape excellent quality images when compared to the camcorder used in the previous experiment. Recording on tape offers good mobility, and an outdoor scene was shot. The tape was edited and converted into individual frames afterwards.

The chosen scene is the interior square of the Whitendale Halls of residence of the University of Central Lancashire on a sunny day (this was a challenge). The camera was placed in the centre of the square. From this position, one can see the front of the building surrounding the square except for the entrance to the square, where the spire of the impressive St. Walburge's church can be seen in the distance.

The camera was panned to the right in a complete circle ensuring that the roof of the building was covered by the field of view. It was then tilted down and panned again in another complete circle, ensuring this time enough overlapping between the two bands.

Most of the scene consists of the building fronts, which lie at about the same distance from the location of the camera, except for the entrance containing the spire in the distance. Motion parallax is found when objects translate at different speeds due to their relative distance to the camera; therefore, very little parallax is present between images except for those containing the spire. Furthermore, since the building front is considerably farther from the camera compared to the amount of involuntary translation, there is little motion parallax in the sequence.

To reduce motion blur, the camera was panned slowly. This yields to a large number of recorded images and very small displacements between successive images (only a few pixels). The high number of images recorded led to a prohibitive computational cost and memory requirement. It was decided to use every tenth frame in the video sequence after verifying that the overlapping between every ten frames was enough to guarantee a successful registration. Finally, 189 images were used to compose the mosaic, each with a resolution of 368×280 pixels after the elimination of one field and horizontal resampling.

The illumination range of the scene was very wide, and considerable contrast was present, particularly in images containing the very bright sky (remember it was a sunny day) neighbouring a shaded building front. This causes dramatic intensity changes in the mosaic that, without correction through image blending, represents a detriment in the quality.

6.4.2 Physical Model for Image Alignment

In this simulation, the images are initially placed in a plane according to successive image alignments. At the end of the simulation, the images will be distributed in three-dimensional space on the surface of a sphere. Since the camera was panned twice at two different tilt angles covering only a limited vertical field of view, there is not enough data to complete an omnidirectional view, i.e. including the sky and the ground. Instead, the images will fall within a band on the surface of a sphere.

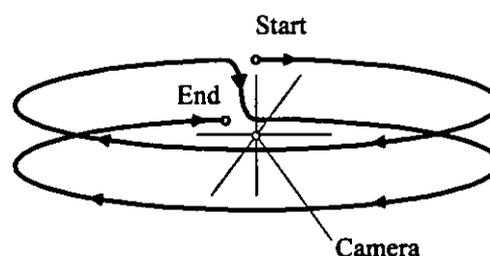


Figure 6-4 Path followed by the camera axis.

The list of neighbour images is initialised with successive image pairs. Two extra neighbour pairs were added manually to the list in order to join the final image of each

band with the first. Three additional pairs at the cardinal points were used to link the two bands.

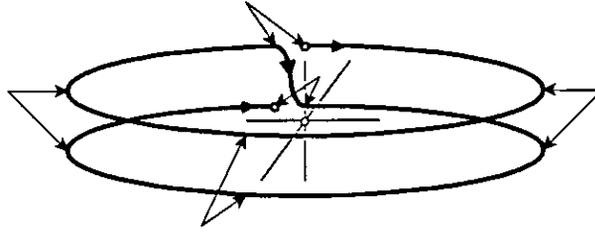


Figure 6-5 The pairs of arrows indicate the extra neighbour images added manually to the list.

The forces of the *hinge constraint* (explained in section 4.4.4) are the main propelling forces which drive the system to the state where all images are at low energy positions with respect to their neighbours.

The hinge constraint alone does not encourage a spherical distribution of the images. As an example with synthetic data (of a perfect single band panoramic loop), Figure 6-6 shows one of the many possible distributions which satisfies the hinge constraint. The image planes are colour coded.

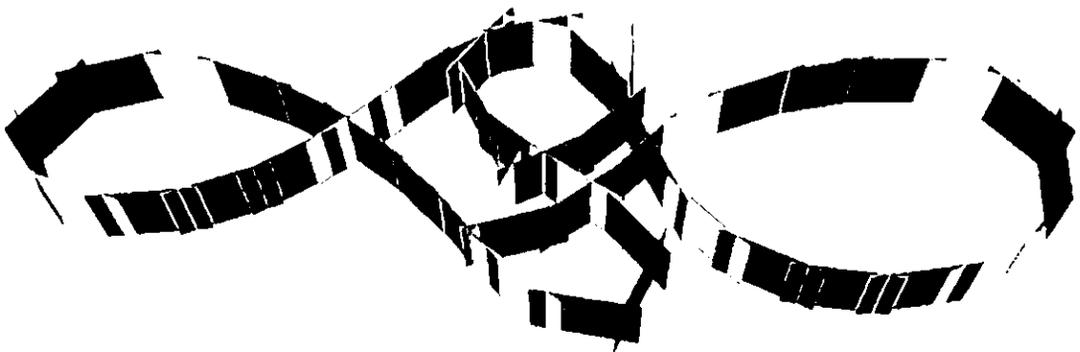


Figure 6-6 One of the possible distributions when only the hinge constraint is present.

In addition to the hinge constraint, two other constraints have been investigated. The *spherical constraint* and the *same plane constraint*.

6.4.2.1 The Spherical Constraint

The *spherical constraint* was designed to induce the spherical distribution of images in the system without having to provide information regarding the radius of the sphere. The spherical constraint effectively makes the images face the same point, which, for convenience, is chosen to be the centre of mass of the system (\mathbf{CM}_{SYS}). The constraint is implemented with two forces of the same magnitude and opposite direction (see Figure 6-7). The points of action of the forces are \mathbf{P}_{CM1} and \mathbf{P}_{CM2} , which are situated at a fixed distance in front and behind the centre of the image respectively.

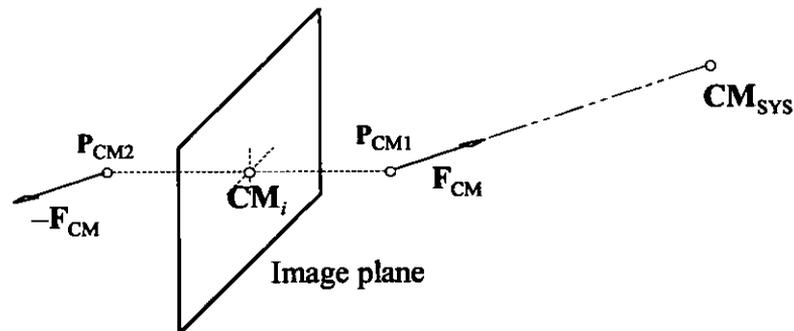


Figure 6-7 Spherical constraint. Auxiliary forces acting on \mathbf{P}_{CM1} and \mathbf{P}_{CM2} rotate the image towards the centre of mass of the system.

The magnitude of the force F_{CM} is proportional to the distance between the centre of the image and \mathbf{CM}_{SYS} . The constant of proportionality is k_{CM} .

$$\mathbf{F}_{\text{CM}} = k_{\text{CM}} (\mathbf{CM}_{\text{SYS}} - \mathbf{CM}_i) \quad (66)$$

The points \mathbf{P}_{CM1} and \mathbf{P}_{CM2} have been arbitrarily located at a distance of 100 pixels from the image plane.

If the spherical constraint is added to every image in the example in Figure 6-6, it will be shown that the new forces can drive the system into a cylindrical distribution. This is shown in Figure 6-8.

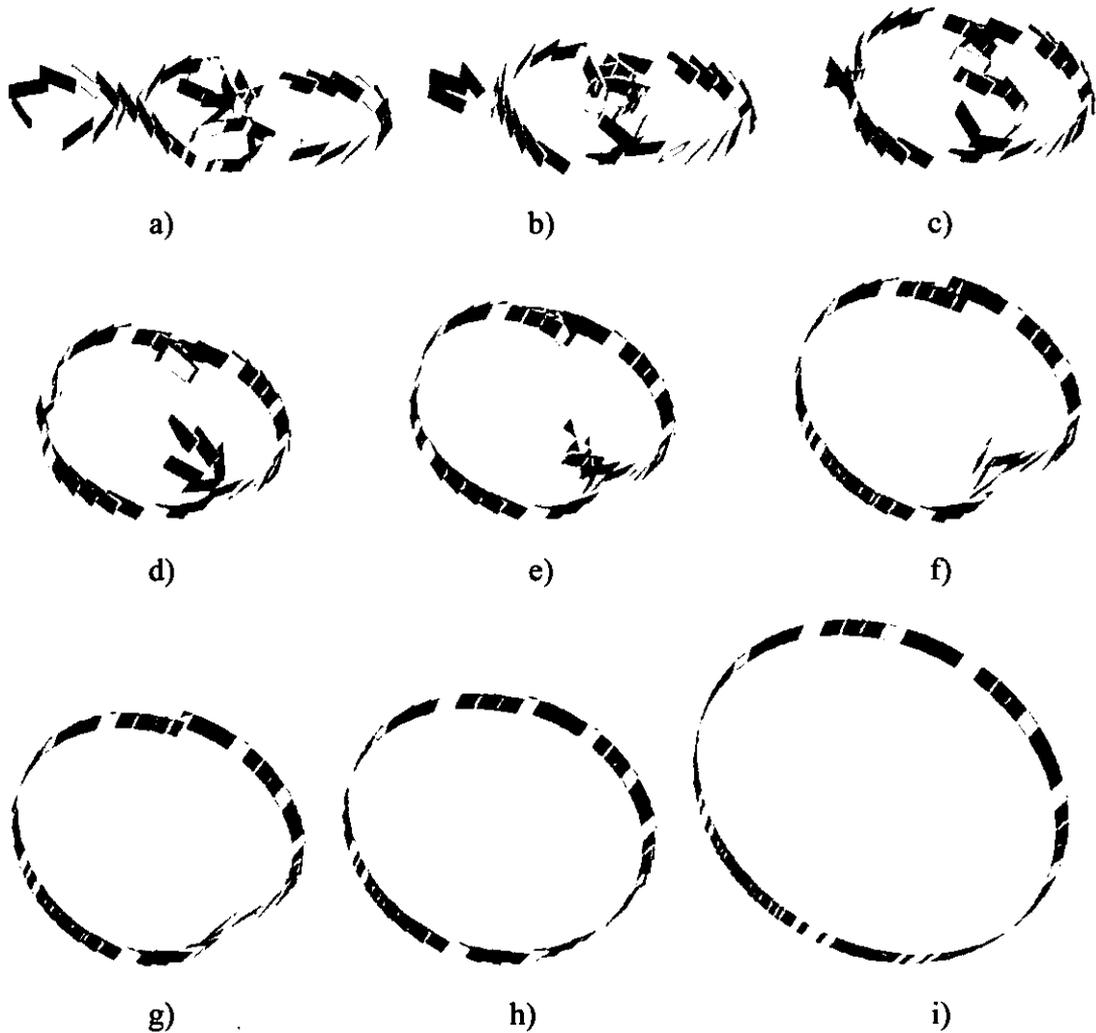


Figure 6-8 Sequence showing evolution of the image planes represented in Figure 6-6 after introduction of the spherical constraint.

However, the value of the constant k_{CM} must be chosen carefully. It is not rare that a double loop or a figure of eight develops from the initial position in the plane (see Figure 6-9). A high value of k_{CM} (with respect to the hinge constant k_H described in section 4.4.4) would prevent the images from turning away from CM_{SYS} , consequently the loop cannot be 'untied', resulting in the stable distribution shown in Figure 6-10. If, on the contrary, the value of k_{CM} is too low with respect to k_H , the progression to disentangle the loop becomes very slow and eventually halts completely (see Figure 6-11). The

appropriate value of k_{CM} was found to be in the order of $0.02k_H$ to $0.1k_H$. Figure 6-12 shows the evolution of the images for $k_{CM}=0.1k_H$.

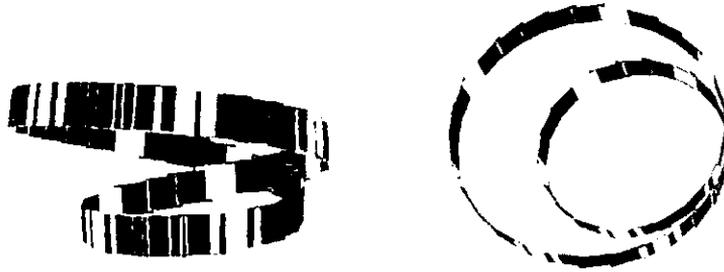


Figure 6-9 A loop can develop from the initial position of the images. The same distribution is here represented from two different points of view.



Figure 6-10 Final distribution of images from the position shown in Figure 6-9. Typical result for a high value of k_{CM} ($k_{CM}=k_H$).



Figure 6-11 Final distribution of images from the position shown in Figure 6-9. Typical result for a low value of k_{CM} ($k_{CM}=0.001k_H$).

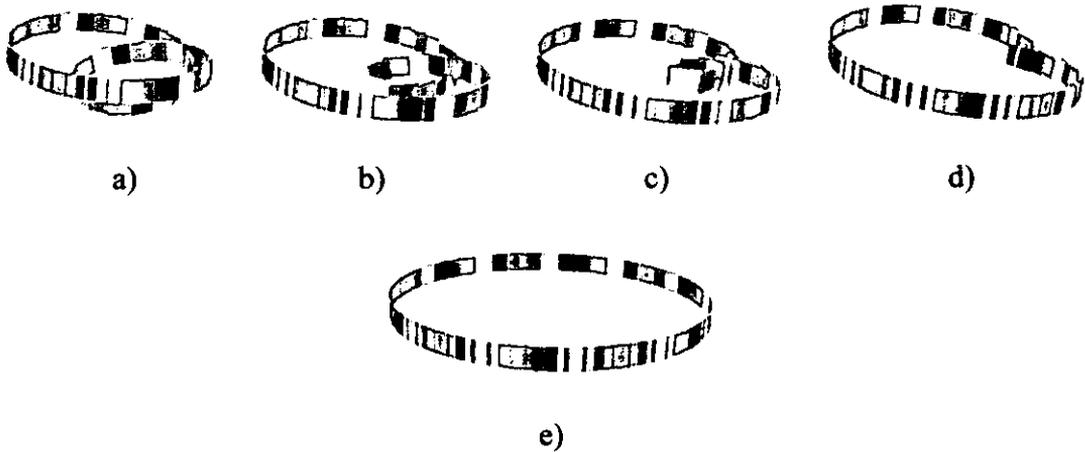


Figure 6-12 Sequence showing the evolution of the images from the position shown in Figure 6-9 (for $k_{CM}=0.1k_H$).

The position of the system's centre of mass is computed supposing the mass of each image is concentrated at its geometrical centre, which lies on the image plane.

$$\mathbf{CM}_{SYS} = \frac{1}{N} \sum_{i=1}^N \mathbf{CM}_i \quad (67)$$

The results obtained with this model are satisfactory. However, if the images are unevenly distributed having a concentration of images at one side of \mathbf{CM}_{SYS} , the position of \mathbf{CM}_{SYS} will not be in the centre of the sphere. For instance, an incomplete panorama (e.g. panning the camera 90°) cannot be composed without information regarding the focal length of the camera, that is, the radius of the sphere.

A failed attempt was made to reduce the effect of misplaced \mathbf{CM}_{SYS} due to differences in image density. The position of the centre of mass for each image \mathbf{CM}_i was displaced from its geometrical centre in the computation of \mathbf{CM}_{SYS} (67). Instead, it was located at a distance in front of the image plane. The distance initially tried was the average distance of all images to \mathbf{CM}_{SYS} , and a later attempt was made using the minimum of these distances. In both cases the new position of \mathbf{CM}_{SYS} diverged away from the centre of the sphere.

The spherical constraint drives the images to a spherical distribution. When equilibrium is reached, the images are distributed approximately on the surface of a sphere. Actually

this shape is not purely spherical but slightly oval. The effects of this are discussed in section 6.5.2.

If the spherical constraint is removed at this point from the model, the images will rearrange to satisfy the hinge constraint only, thus reducing further the remaining error. This final rearrangement must be overdamped or critically damped to prevent images from distorting the spherical shape more than necessary. Hence, the images will not lay exactly on the surface of a sphere but at a position that reduces further the visible error in the mosaic.

6.4.2.2 The Same Plane Constraint

In another unsuccessful effort to encourage a spherical distribution of the images, the *Same Plane Constraint* was developed. This constraint pulls two neighbour images toward the same plane, i.e. the angle between the image planes is encouraged towards zero. The constraint is implemented using a second non-rigid hinge (see Figure 6-13). This hinge is defined, for every pair of overlapping images, by two points in one image (P_{S1A} and P_{S2A}) which are attracted to the corresponding two points in the other image (P_{S1B} and P_{S2B}). This attraction is a force whose magnitude is proportional to the distance between corresponding points.

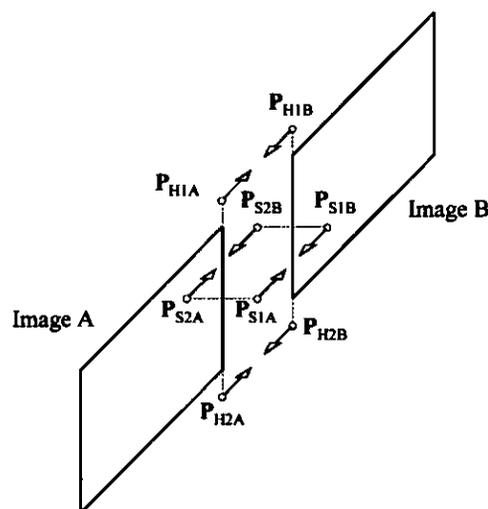


Figure 6-13 Same Plane constraint. Auxiliary forces acting on P_{S1A} , P_{S1B} , P_{S2A} and P_{S2B} pull the images towards the same plane.

The intention was to push every pair of images towards a low inter-plane angle, since the angle between neighbour image planes is often very small. However, when a loop develops, such as the one depicted in Figure 6-9, the smaller loop grows while the outer shrinks, and the system converges to two overlapping circles.

Even in the absence of a loop, when the constraint was applied to the image set, the two image bands (corresponding to the two 360° camera pans) were distributed on the surface of a cylinder rather than on the sphere. That is, the angle between images on the upper band with respect to the images in the lower band was zero. This introduced unacceptable distortions. The same plane constraint was therefore not used in the final system.

6.4.2.3 Hinge Parameters

The proposed hinge constraint uses forces proportional to the distance between corresponding hinge points, i.e. springs. Since these forces are linear, the value of the constant of proportionality k_H in equation (40) does not have any effect on the final distribution of the images (if no other constraint is used simultaneously), arriving always at the same minimum error distribution.

Non-linear forces have also been tried, in particular, forces proportional to the square of the distance between corresponding hinge points.

$$\begin{aligned} F_1 &= k'_H (\mathbf{P}'_{H1B} - \mathbf{P}'_{H1A})^2 \\ F_2 &= k'_H (\mathbf{P}'_{H2B} - \mathbf{P}'_{H2A})^2 \end{aligned} \quad (68)$$

With non-linear forces, the system can arrive at different equilibrium positions which depend on the constant k'_H . For each of them, the maximum error and the average error given by equations (57) and (58) are different, and no simple relationship between the value of k'_H and the magnitude error has been found. The system seems to find a different local minimum energy for different values of k'_H .

Experimental results using linear and non-linear forces show that the visual differences between the two is negligible. A series of mosaics were created using different

combinations of options. Table 6-1 shows the error at the equilibrium position of four significant experiments.

The maximum error (57) and the average error (58) are based on the distance between hinge points and are a quantitative measure of the quality of the mosaic. The column "Quality" shows a subjective visual assessment on the resulting mosaic.

Experiments *iii* and *iv* are continued from *i* and *ii* respectively by releasing the spherical constraint. There is a significant improvement when the spherical constraint is released.

The linear and non-linear approach differ in the distribution of the error. In the linear approach, the maximum error is allowed to grow for the benefit of the average error. One pair of neighbour images may be left slightly misaligned to allow the rest to be better aligned. In the non-linear approach, the maximum error is kept to a lower value by increasing the average error.

Experiment	k'_H	k_H	k_{CM}	Max. Error (pixels)	Aver. Error (pixels)	Quality
<i>i</i>	0	10	1	3.416	0.709	<i>poor</i>
<i>ii</i>	10	0	1	3.286	1.014	<i>poor</i>
<i>iii</i>	0	10	(released)	2.912	0.509	<i>good</i>
<i>iv</i>	10	0	(released)	2.276	0.571	<i>good</i>

Table 6-1

6.4.2.4 Effects of Damping on Convergence

A damping force is necessary to dissipate the excess energy of the system and to prevent it from oscillating indefinitely. For each image, a damping force and a damping torque, proportional to the square of the velocity and to the angular velocity respectively, are introduced in the model.

The effect is that of submerging the system in a viscous fluid. The images are free to move, but they do not develop excessive speed, nor do they oscillate as much. By carefully choosing the damping constant, the system can be made underdamped, overdamped or critically damped, with different times to equilibrium for each case.

It was observed that near equilibrium, when the velocities are small, the dissipation of the remaining energy was slow, taking a long time to stop the low amplitude oscillations. This problem was solved by making the drag force proportional to the velocity as well (69). In fact, a more precise model of viscous drag has a linear component:

$$F_{\text{drag}} = k'_d V^2 + k_d V \quad (69)$$

6.4.2.5 Voronoi Tessellation and Delaunay Diagram on the Sphere

After equilibrium is reached, in order to identify new neighbours, Delaunay triangulation (section 3.4) is used. In the spherical mosaic, the images are distributed on the surface of a sphere. The implementation of Delaunay triangulation developed for the planar mosaic requires, however, the 2D positions of the centres of the images.

Thus, the identification of neighbour images of a given image i is performed by projecting the centres of the candidate images (those close enough to be neighbours) on the plane of image i . The neighbours are given by Delaunay triangulation on these points in the plane given by image i .

6.4.3 Image Integration

At this stage, each image has a determined position and orientation in the three-dimensional space. In this experiment, the resulting mosaic can be seen as a panoramic image due to its limited vertical field of view. Thus, this mosaic can be represented with a single image projected on a cylinder (see Figure 3-5).

For each pixel in the cylindrical image, a ray will be issued from CM_{SYS} that passes through the pixel. This ray will intersect with one or more images. From the intersected

image nearest to CM_{SYS} , the pixel value at the intersection of the ray with this image will be copied to the pixel in the cylindrical image.

6.4.4 Results

Figure 6-15 a) and b) show the resulting mosaic; in a) the images are colour coded to show the tessellation, i.e. the individual contribution of each image. The mosaic has a very high aspect ratio with a resolution of 4096×512 pixels. In order to view it without distortion, the image should be pasted on a cylinder, and the viewer placed in its centre.

The difference in brightness between some neighbour images makes the seams visible, and reduces the quality of the mosaic. This is particularly noticeable in the shaded front of the building (shown in Figure 6-14).

Note how the lower band and the upper band have been aligned correctly. Some distortion was expected at the spire due to motion parallax, however, the results are excellent even at locations with less accurate registrations.



Figure 6-14 Details of mosaic in Figure 6-15b.

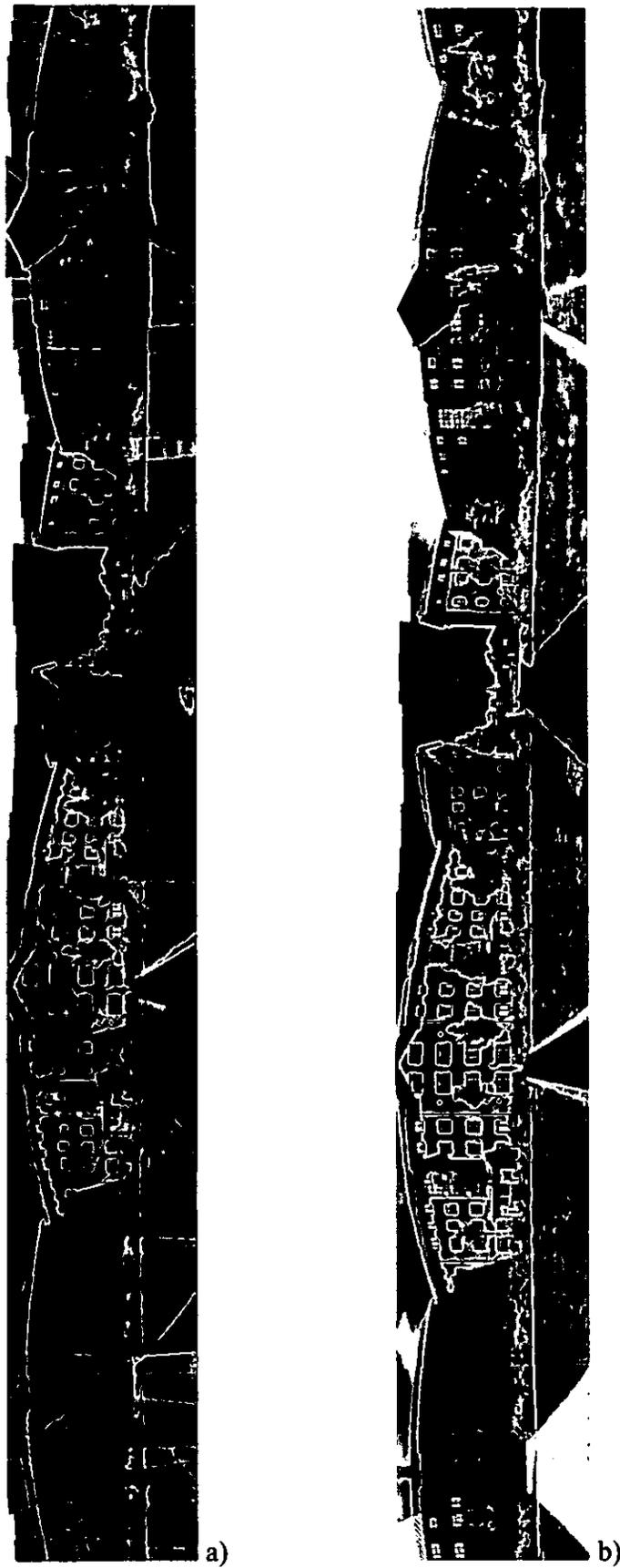


Figure 6-15 The complete mosaic showing in a) the contributions of the individual images in different colours.

6.5 Full Spherical Mosaic

A final experiment was conducted to demonstrate the proposed mosaic construction method. In this section, the experiment on the full spherical mosaic is described.

6.5.1 Image Acquisition

The images in this experiment were obtained using a still digital camera. The camera was connected to a computer where the images were downloaded. Although it is a time consuming process when the number of images is large, the still camera offers more accurate control over the number of images, the amount of overlap and the path to follow.

The scene is Room 111 in Kirkham Building at the University of Central Lancashire, where the entire project has been realised. The camera was mounted on a tripod that enabled the camera to rotate about a point sufficiently close to its optical centre to obtain parallax-free images. The camera was panned and tilted by small angles between shoots. The path followed was carefully planned in advance according to the following requirements:

- i. To ensure that the images cover the entire omnidirectional view, i.e. without leaving gaps in the mosaic.
- ii. To provide sufficient overlapping between adjacent sections of the path for accurate registration.
- iii. To avoid redundant images, i.e. too many images from the same regions of the scene.
- iv. To ensure that sufficient texture is present in every pair of neighbour images, several small yellow notes were laid on the carpet to provide a reference for the registration, and a sign reading “HELLO” was stuck to the ceiling.
- v. To avoid objects too close to the camera that may appear defocused.

The camera exposure time was set to the same constant value for all the images. This explains why the fluorescent lamps are too bright in the images of the ceiling. This was necessary in order to obtain the dark regions of the scene (e.g. the red carpet) with a suitable brightness.

The trajectory of the camera is represented in Figure 6-16. Images of the floor directly under the camera could not be taken.

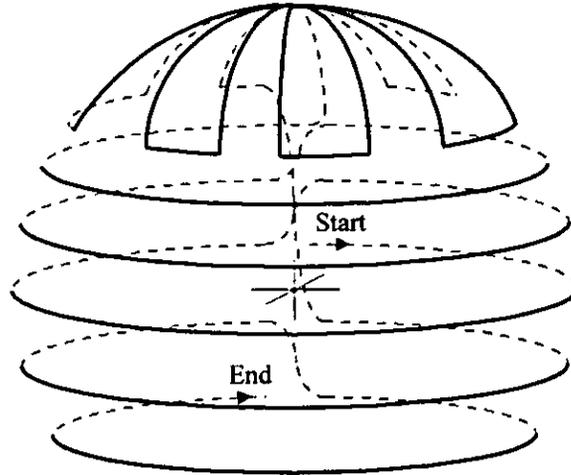


Figure 6-16 Trajectory of the camera.

6.5.2 Physical Model for Image Alignment

The physical model for this experiment has been described fully in section 6.4.2. In the initialisation of the neighbour list, 24 extra neighbour pairs were included in the list in order to link both the bands and the images at the “HELLO” sign on the ceiling. After the images were settled in their equilibrium positions, the number of neighbour images was calculated using Delaunay triangulation. The list of neighbours was updated with a total 1254 of pairs.

The images are distributed approximately on the surface of a sphere. The sphere would be complete except for the missing images under the camera.

Figure 6-17 to Figure 6-22 show the evolution towards equilibrium of the system of images initially placed sequentially on a plane. The constant values chosen for this simulation were $k'_H=0$, $k_H=10$, $k_{CM}=0.3$, $\Delta t=0.025$ and the spherical constraint was

released to obtain the final positions. The maximum error was 7.352 pixels and the average error was 0.962 pixels.

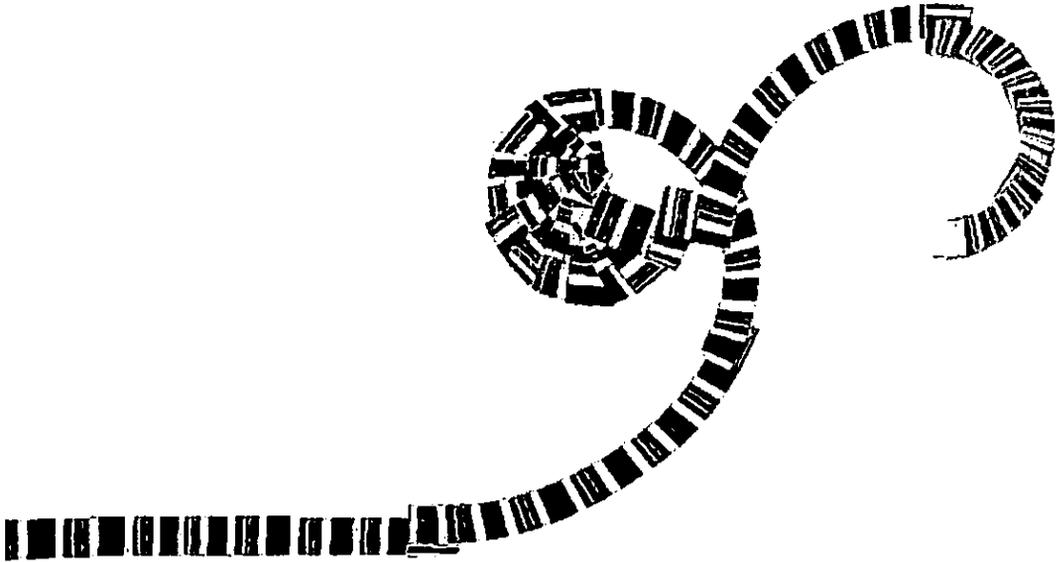


Figure 6-17 Initial position with images placed sequentially on a plane. The path followed by the camera over the scene is discernible in this figure. From the left, the first straight section is the first 360° pan at horizontal level. The 15° tilt up can be seen at the end of it. Next, the curved section is the next 360° pan and so on.



Figure 6-18 After 25 iterations (i.e. time steps), the images have moved towards the low energy positions with respect to their neighbours.

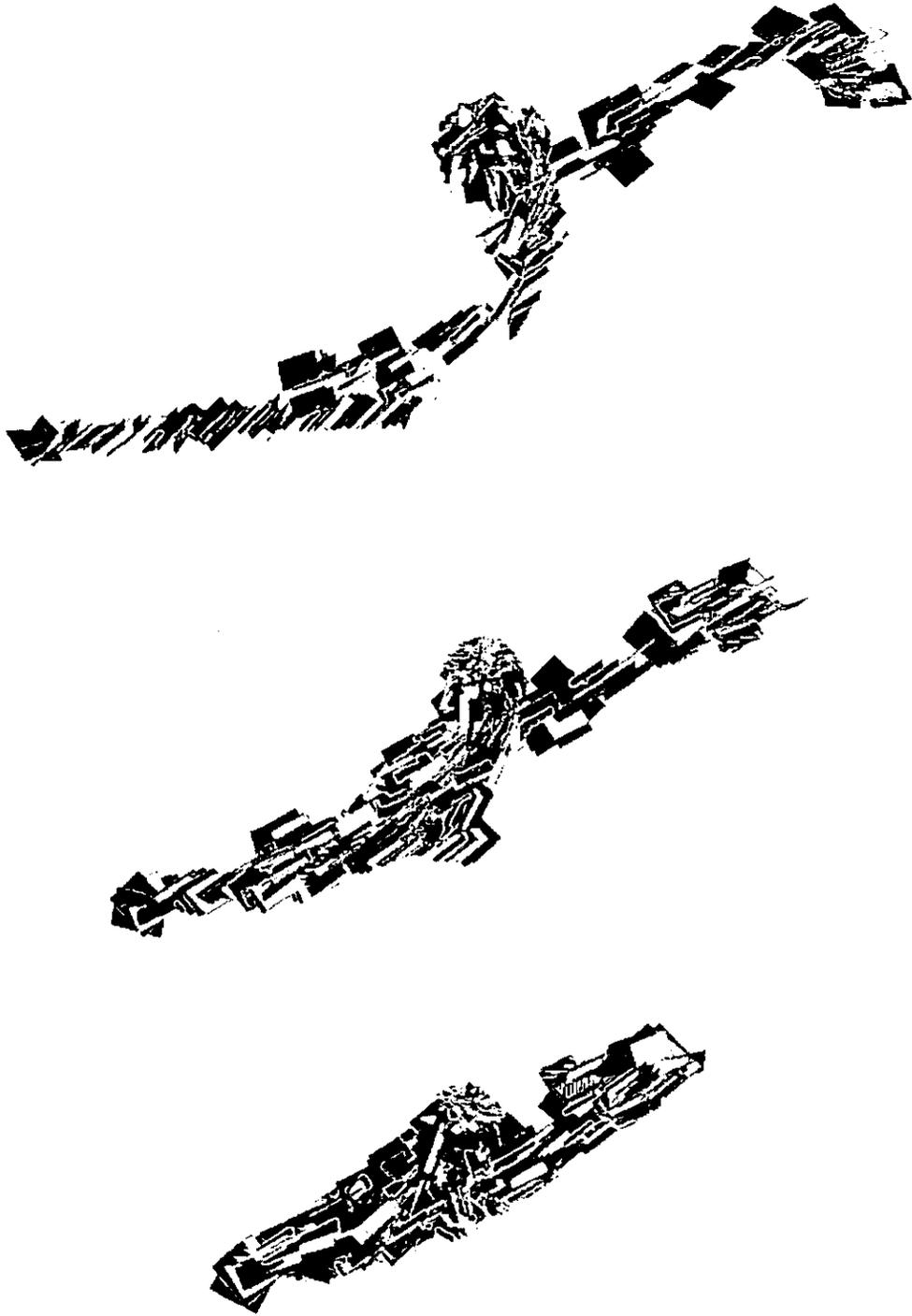


Figure 6-19 State of the simulation at iterations 50, 100 and 150.

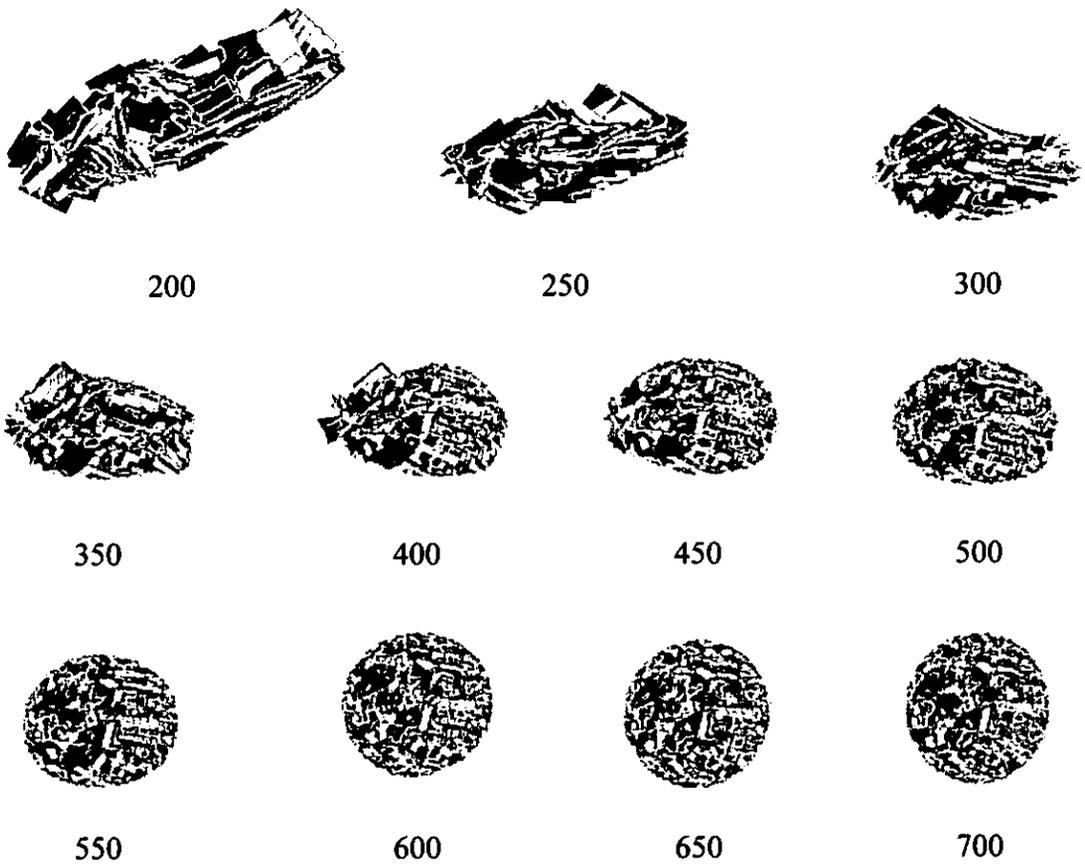


Figure 6-20 From iteration 200 to 700, showing the evolution of the images towards a spherical shape.

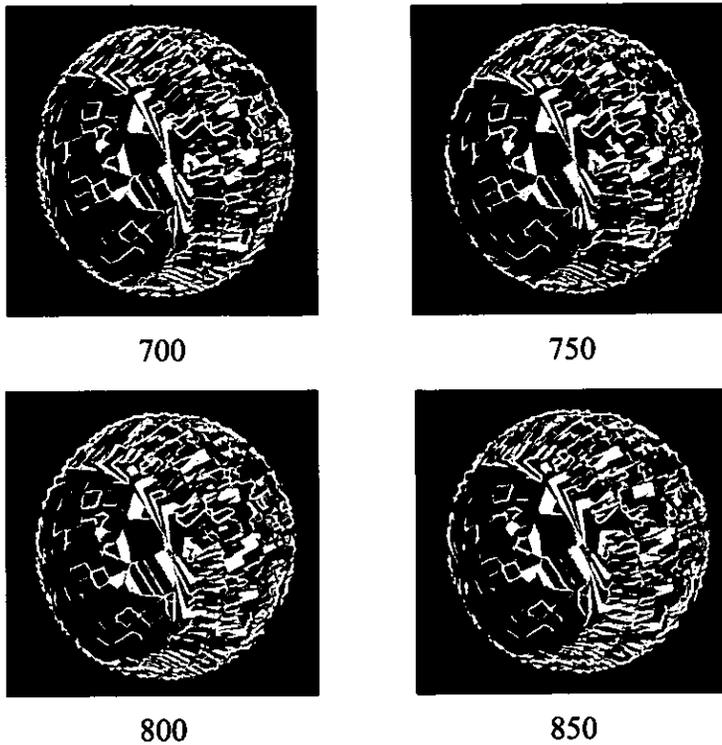


Figure 6-21 This representation shows distinctively the shape of the image distribution at iterations 700 and following.

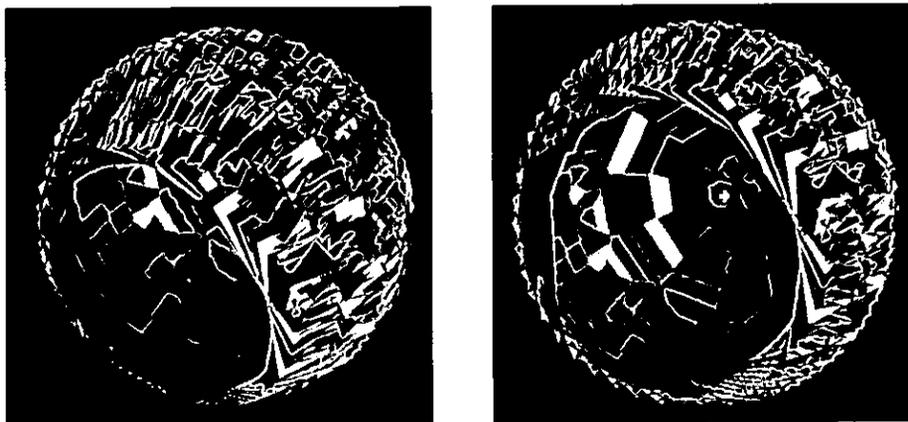


Figure 6-22 Two views of the final shape at iteration 900. The small white sphere in the centre of the right image represents the position of CM_{SYS} .

The fact that the sphere is incomplete at the bottom and that there is a concentration of images at the ceiling, affects the position of CM_{SYS} . When viewing the mosaic from this displaced CM_{SYS} , this will have the effect of slightly bending the straight lines in the scene.

The simulation was run on a Pentium 166 MHz, and it took about 7 minutes to complete. This time can be reduced if the images are not displayed during the simulation.

6.5.3 Image Integration and Display

Once the position of the images are known, it is necessary to find a practical means of representing the mosaic which can be quickly rendered or viewed.

A traditional way to do this is with a spherical map which can be rendered by applying it as a texture on a spherical mesh. However, the resolution of such texture is unevenly distributed requiring more pixels at the equator than at the poles. The rectangular texture would have an inefficient distribution of resolution when mapped on the sphere.

[Szeliski and Heung-Yeung, 1997] suggested mapping the mosaic onto an arbitrary texture-mapped polyhedron surrounding the origin, this way the virtual environment can be explored using standard 3D graphic viewers and hardware without requiring special-purpose players. This choice can range from something as simple as a cube with six separate texture maps [Greene, 1986], to something as complicated as a subdivided dodecahedron, or even a latitude-longitude tessellated globe.

A cube with six separate texture maps was chosen for simplicity and for efficiency when using standard 3D graphic rendering engines.

For each side of the cube, for each pixel in the texture image, a ray is traced from CM_{SYS} through that pixel. From the intersected image nearest to CM_{SYS} , the pixel value at the intersection of the ray with this image will be copied to the pixel in the texture image.

The seams between adjacent images were blended using a weighted average of the pixels near to the seam. Although simple, this improves considerably the perceived quality of the mosaic. Figure 6-23 shows the effect of the image blending used in this experiments.



Figure 6-23 After blending the seams, a smooth transition between the images helps to hide small misalignments.

6.5.4 Displaying Results

Figure 6-24 shows the six sides of the cube where each image is represented in a different colour to show the contribution of each of them. The region in red indicates the position, not covered by any image, where the tripod holding the camera was placed. Figure 6-25 shows the six images ready to be rendered on the interior surface of a cube.

Once the textures to be mapped on the cube have been constructed, they can be rendered with a standard 3D graphic system.

A model in the VRML (Virtual Reality Modelling Language) was written containing the geometry of the cube and the references to the texture image files. VRML is a standard 3D graphic description language used extensively in the internet today. An internet browser with a VRML plug-in can display the resulting mosaic as shown in Figure 6-27.

The user is virtually placed in the centre of the cube, and can look in any direction. VRML allows linking of selected areas on the mosaic to other web pages or other spherical mosaics, offering the possibility to create virtual scenarios with linked omnidirectional views.

6.6 Summary

This chapter has illustrated the evolution of the research through three significant experiments, fulfilling the title of this thesis in the accomplishment of the spherical mosaic.

The construction of the spherical mosaic has been demonstrated as the proof of the validity of the methods proposed in this thesis regarding the use of a physical simulation for consistent image alignment.

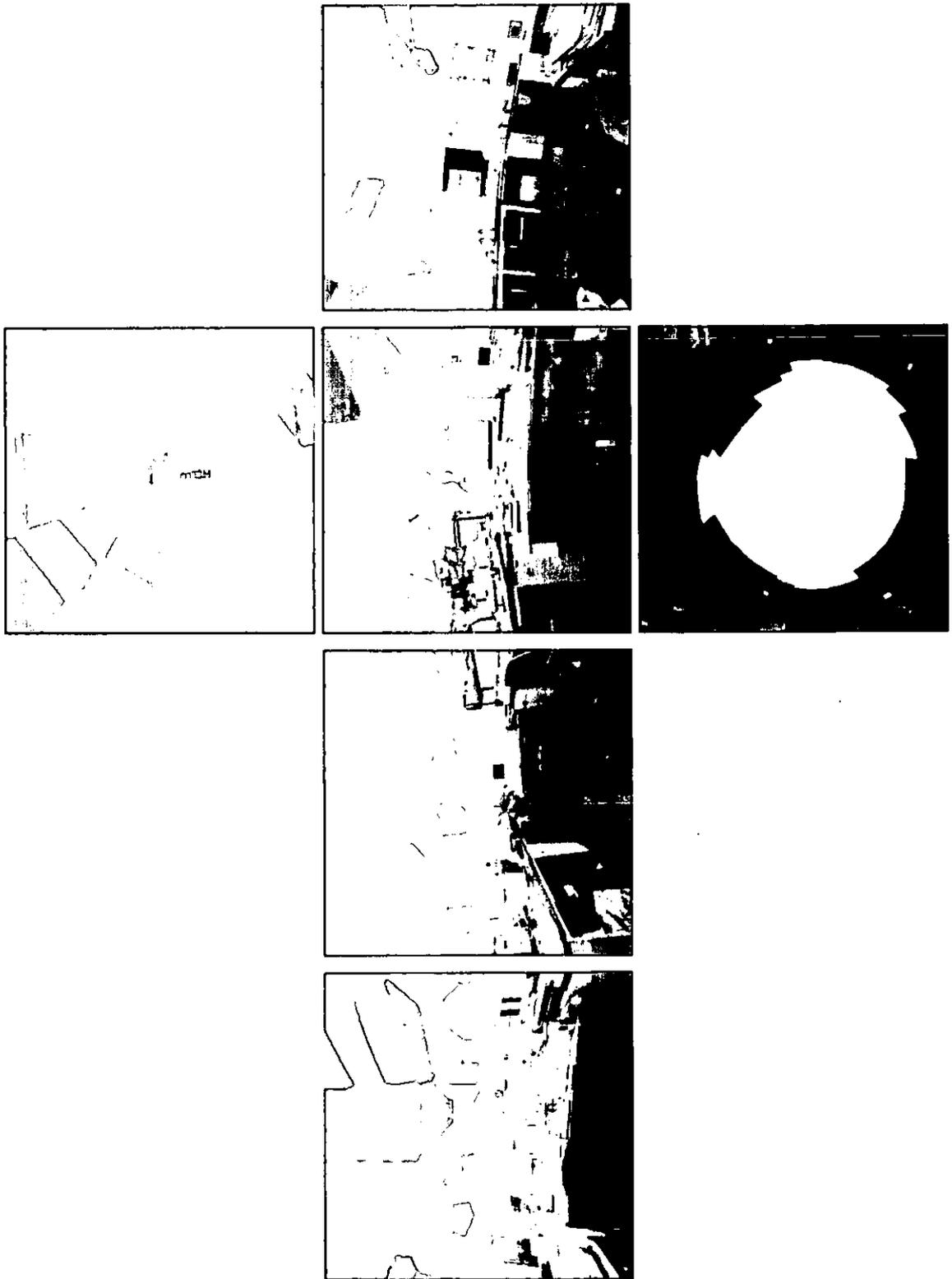


Figure 6-24 The six sides of the cube showing the contribution of each image on the mosaic in different colours.

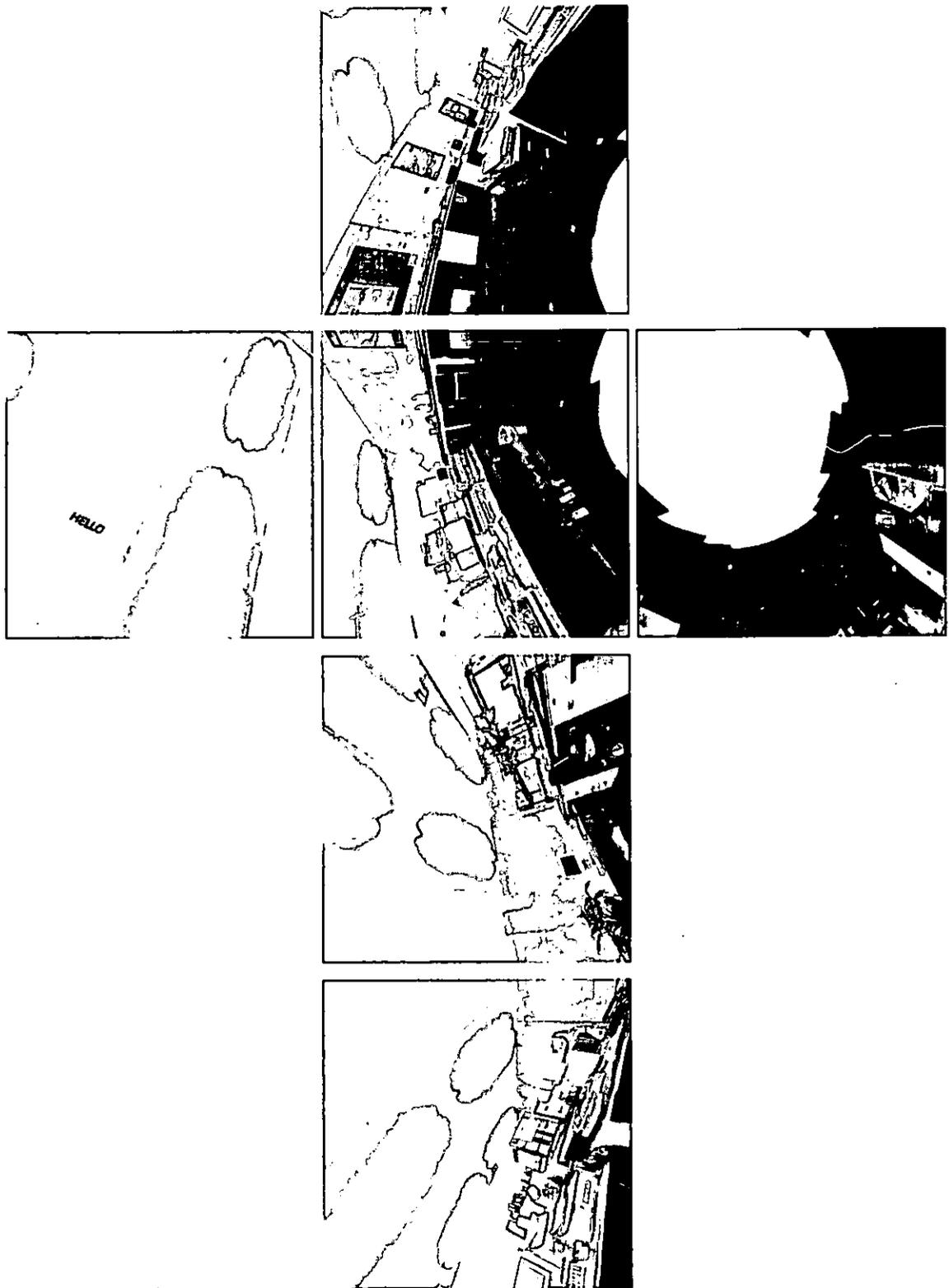


Figure 6-25 The six sides of the cube ready to be rendered.

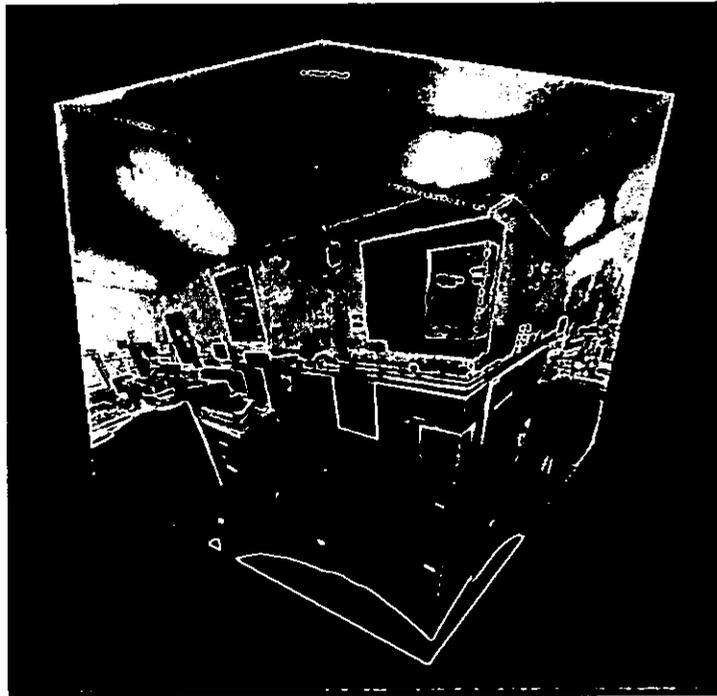


Figure 6-26 The cube as seen from outside.



Figure 6-27 An internet browser (Netscape) with a VRML plug-in (Cosmo Player) are used to view the spherical mosaic.

Chapter 7 CONCLUSIONS & FURTHER WORK

7.1 Contributions

The research reported in this thesis has been dedicated to the problem of mosaic construction, in particular to the cases where consistent image alignment is necessary. When only successive images are registered, mosaics with looping paths can present significant misalignments due to the accumulation of small errors from the individual registrations. The main objective of this research was to develop novel schemes for the elimination of visual defects in mosaics due to error accumulation. In this final chapter of the thesis, a summary of the original contributions is presented. Research areas in which future work would be appropriate are also discussed.

Summarised descriptions of the contributions are:

1. Section 4.4.3 introduced the physical model for the images, the idea of associating an image to a position and orientation in three-dimensional space allows representation of their spatial relationship without the need to project them onto a particular surface.

Furthermore, the information extracted by registration of a pair of overlapping images, which is usually represented by a projective transformation matrix, can be seen in a more graphical and intuitive way as the position and orientation of the images in space.

2. By using the ideal camera model, where the scene is projected on the sensor plane without distortions, it has been established that the relative position and orientation of two images corresponds to the position and orientation of the camera's sensor plane at the time when the images were taken.

By investigating the motion of the sensor plane new relationships have been identified between images obtained by a rotating camera. The hinge model is based on the intersection line between the camera sensor plane at different times. The intersection line has the property that the images can be seamlessly joined through that line.

3. The looping path problem has been identified which had not previously been satisfactorily solved. Most of the research into mosaic construction has been oriented towards real time systems, where the images arrive in a timely fashion to the processor and the mosaic must be updated with each new image.

Today's technology offers high storage and processing capabilities, enabling the storage of the images for off-line processing. New applications are now possible with the availability of desktop cameras.

4. The distribution of the small alignment errors has been demonstrated through several experiments as a practical solution to the looping path problem. Each image is allowed to modify its position relative to its neighbour images in the mosaic by a small amount with respect to the computed registration. These small modifications allow for a greater gain in the quality of the mosaic. This solution is independent of the registration technique used to align individual pairs of images, hence a suitable implementation can be used for different types of images.
5. The major contribution of this thesis is the use of an analogy between the images and a physical model. The images correspond to rigid objects, and these are linked with forces which pull them towards the correct positions with respect to their neighbours. This network of self-organising images is implemented in a computer simulation.

The familiarity with the physical world allows the researcher to create meaningful relationships between the images. This has been demonstrated with the design of two constraints for the spherical mosaic, the hinge constraint and the spherical

constraint. The study of the convergence of the system and the validation of hypothesis have been possible by interacting with the model during the simulation.

As a direct result of the new concepts developed in this research work, a spherical mosaic was constructed. Where others have used omnidirectional cameras or known geometry arrangements to obtain such views, the creation of mosaics was limited by the looping path problem.

The research described in this thesis has found a solution to the looping path problem and developed a new method of spherical mosaic construction using a physical simulation.

7.2 Further Work

The implemented registration algorithm computes the transformation that best aligns two images. However, it does not provide any information regarding the validity of the result. That is, trying to align two images containing insufficient information may yield an incorrect result. For instance, there were very little features in some pairs of images of the ceiling of the room in the experiment in section 6.5. Some pairs of images were difficult to align even manually due to the lack of content in the overlapping region. Since the content of the scene will vary, an automatic mosaic process should be able to identify these cases. A variable should be computed by a validation algorithm indicating the reliability of the result of the registration.

The information given by this variable could then be used in the physical model to change the spring constants for example, where the attraction force will be less for ambiguous registrations, and stronger for reliable registrations.

The registration algorithm implemented does not account for scale variations between the images, i.e. zoom changes. Zooming in the parts of the scene that contain more detail could be used to selectively increase the resolution of the mosaic.

To model different scale images, the images could shrink or expand according to a scale force to satisfy the hinge constraint. In this model, the hinge points between two images would be set with both images at the same scale as shown in Figure 7-1.

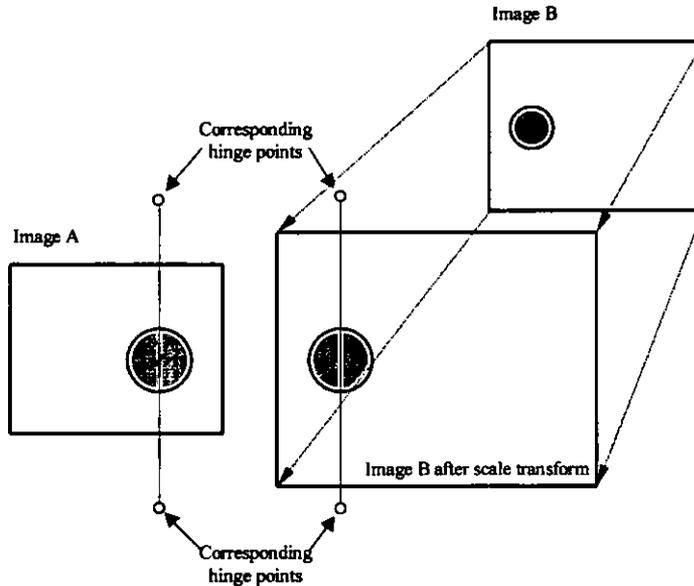


Figure 7-1 Hinge constraint model for scale changes.

Alternatives or modifications to the proposed spherical constraint could be investigated to place the centre of mass of the system, CM_{SYS} , more accurately in the centre of the sphere even when the image density is not constant throughout the sphere.

The separation between the hinge points in the same image has been established arbitrarily at 400 pixels in all the experiments. Although good results have been obtained with this setting and the effect of this parameter has been discussed in section 4.4.4, a more thorough study could be carried out to determine the best value.

Experiments using synthetic data could assist the identification of relationships between the different parameters involved in the mosaic construction process.

An image registration technique based on the principle that two images acquired with a rotating camera share the same pixel intensity through a line could be developed. This line is given by the intersection of the camera's sensor plane at the time the images were

taken. Szeliski and Heung-Yeung [Szeliski and Heung-Yeung, 1997] have proposed a registration method which directly recovers rotations (pan and tilt angles) instead of general 8-parameter perspective transforms. This technique could be optimised for finding the intersection line defined by the hinge constraint.

7.3 Summary

Spherical mosaic construction is now achievable as a direct result of this research work. Others have used omnidirectional cameras or known geometry arrangements to obtain such views, and the creation of mosaics was limited by the looping path problem. This problem has been overcome by using a physical analogy that consistently aligns the images.

REFERENCES

- Adelson EH and Bergen JR, "The Plenoptic Function and the Elements of Early Vision", *M. Landy and J. A. Movshon, editors, Computational Models of Visual Processing*. MIT Press, 1991.
- Alvert L, Zobrist, Nevin A, Bryant A and McLeod RG, "Technology for large digital mosaics of landsat data", *Photometric Engineering and Remote Sensing*, vol. 49, No. 9, pp. 1325-1335, 1983.
- Aurenhammer F, "Voronoi diagrams: a survey of a fundamental geometric data structure", *ACM Computing Surveys*, 23(3):345-405. 1991.
- Baraff D and Witkin A, "Physically Based Modelling: Principles and Practise", *SIGGRAPH'97 Course Notes*. 1997.
- Barnea DI and Silverman HF, "A class of algorithms for fast digital image registration", *IEEE Transactions on Computers*, vol. C-21, No. 2, pp. 179-186, February 1972.
- Batista J, Dias J, Araújo, Traca de Almeida A, "Monoplanar Camera Calibration, Iterative Multi-step Approach", *Proceedings of the British Machine Vision Conference*, pp. 479-488, 1993.
- Be Here, *Be Here Corporation*, <http://www.behere.com>
- Bender W, "Affine Transformations of Video Sequences", *IS&T's 46th Annual Conference*, pp. 50-52, 1993.
- Bergen JR, Anandan P, Hanna KJ and Hingorani R, "Hierarchical Model-based Motion Estimation", *European Conference on Computer Vision*, 1992.
- Berman DF, Bartell JT and Salesin DH, "Multiresolution Painting and Compositing", *Computer Graphics Proceeding*, pp. 85-90, 1994.
- Brown LG, "A survey of image registration techniques", *ACM Computing surveys*, 24(4):325-376, December 1992.
- Burt P and Adelson EH, "A Multiresolution Spline with Application to Image Mosaics", *ACM Trans. on Graphics*, vol. 2, pp. 217-236, 1983a.
- Burt P and Adelson EH, "The Laplacian Pyramid as a Compact Image Code", *IEEE Trans. on Communication*, vol. Com-31, pp. 532-540, 1983b.

- Burt P and Anandan P, "Image Stabilization by Registration to a Reference Mosaic", *ARPA Image Understanding Workshop*, pages 457-465, Monterey, California, Morgan Kaufmann, November 1994.
- Burt P, Hansen M and Anandan P, "Video Mosaic Displays", *Proceedings of SPIE - International Society for Optical Engineering*, Vol. 2736, pp.119-127, 1996.
- Capel D and Zisserman A, "Automated Mosaicing with Super-resolution Zoom", *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Santa Barbara, 1998.
- Cheeseman P, Kanefsky B, Kraft R and Stutz J, "Super-resolved Surface Reconstruction from Multiple Images", *Technical Report, NASA*, 1994.
- Chen S and Williams L, "View interpolation for image synthesis", *Computer Graphics (SIGGRAPH' 93)*, pp. 279-288, August 1993.
- Chen SE, "QuickTime VR - An image-based approach to virtual environment navigation", *Proceedings of Siggraph' 95*, ACM, New York, pp. 39-46, 1995.
- Delaunay B, "Sur la Sphere Vide", *Bulletin of Academy of Sciences of the USSR*, pp. 793-800, 1934.
- Dirichlet GL, "Über die Reduction der Positiven Quadratischen Formen mit Drei Unbestimmten Ganzen Zalen", *J. Reine u. Angew. Math.*, 40:209-227, 1850.
- Eccles S, "Panorama Creation", *Digit 7*, pp. 86-87, January 1999.
- Fischler MA and Bolles RC, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated cartography", *Communications of the ACM*, 24(6):381-395, 1981.
- Fitzgibbon AW and Zisserman A, "Automatic Camera Recovery for Closed or Open Image Sequences", *Proceedings of European Conference on Computer Vision*, pages 311-326, 1998.
- Giaccone PR, Greenhill D and Jones GA, "Segmenting, Describing and Compositing Video Sequences Containing Multiple Moving Elements", *TV and Broadcasting on Internet, WWW and Networks*, Bradford, 22-23 April 1998.
- Gold R, "Ubiquitous computing and augmented reality", *Computer Graphics (SIGGRAPH' 93)*, pp. 393- 394, August 1993.
- Greene N and Heckbert P, "Creating raster Omnimax images from multiple perspective views using the elliptical weighted average filter", *IEEE Computer Graphics and Applications*, 6(6): 21-27, June 1986.
- Greene N, "Environment mapping and other applications of world projections", *IEEE Computer Graphics and Applications*, 6(11): 21-29, November 1986.

- Guillén M, Holifield P and Varley M, "Improved Mosaic Construction by Accumulated Alignment Error Distribution", *Proceedings of the British Machine Vision Conference*, Vol. 2, pp. 377-387, 1998.
- Guillén M, Holifield P and Varley M, "Spherical Mosaic Construction using Physical Simulation for Consistent Image Alignment", *Proceedings of the Seventh International Conference on Image Processing and its Applications*, Vol. 1, pp. 311-315, 1999.
- Gümüstekin S and Hall RW, "Generation of Large field of view mosaic images", *Tech. Report, Univ. of Pittsburgh Department of Electrical Engineering TR-CVPR-96-01*, 1996a.
- Gümüstekin S and Hall RW, "Mosaic Image Generation on a Flattened Gaussian Sphere", *Third IEEE Workshop on Applications on Computer Vision*, Florida, pp. 50-55, December 1996.
- Halfhill TR, "See you around", *Byte Magazine*, pages 85-90, May 1995.
- Hamit F, "360-Degree Interactivity: New Video and Still Cameras Provide a Global Roaming Viewpoint", *Advanced Imaging*, pp. 50-52, March 1997.
- Hansen M, Anandan P, Dana K, van der Wal G and Burt P, "Real-time scene stabilization and mosaic construction", *ARPA Image Understanding Workshop*, pages 457-465, Monterey, California, Morgan Kaufmann, November 1994.
- Hartley R and Gupta R, "Linear pushbroom cameras", *J. Eklundh, editor, Third European Conference on Computer Vision*, pages 555-566, Stockholm, Sweden, Springer, May 1994.
- Hartley RI, "Euclidean Reconstruction from Uncalibrated Views", *J. Mundi, A. Zisserman, and D. Forsyth, editors, Applications of Invariance in Computer Vision*, LNCS 825, pages 237-256. Springer-Verlag, 1994.
- Heckbert PS, "Fundamentals of Texture Mapping and Image Warping", *Master Thesis, Dept. of EECS, UCB*, Technical Report No. UCB/CSD 89/516, June 1989.
- Horii S, Oshima Y and Hirao K, "Digital Image Processing", *Eighteenth International Symposium on Remote Sensing of Environment*, pp. 1785-1794, 1995.
- Hsu CT and Wu JL, "Multiresolution Mosaic", *IEEE Transactions on Consumer Electronics*, Vol. 42, No. 4, November 1996.
- Irani M and Peleg S, "Improving resolution by image registration", *Graphical Models and Imag. Proc.*, May 1991.
- Irani M, "Applications of image mosaics", *International Conference on Computer Vision*, Cambridge, MA. November 1995.
- Irani M, Anandan P, and Hsu S, "Mosaic Based Representations of Video Sequences and Their Applications", *IEEE International Conference on Computer Vision*, pp. 605-611, 1995.

- Janssen RDT and Vossepoel AM, "Compilation Of Mosaics From Separately Scanned Line Drawings", *IEEE Workshop on Applications of Computer Vision - Proceedings*, pp. 36-43, 1994.
- Krishnan A and Ahuja N, "Panoramic image acquisition", *IEEE Conference on Computer Vision and Pattern Recognition*, pages 379-384, San Francisco, California, June 1996.
- Lahart MJ, "Optical area correlation with magnification and rotation", *J. Optical Society Am* 60, No. 3, pp. 319-325, 1970.
- Lim JS, "Two-Dimensional Signal and Image Processing", *Prentice-Hall International Editions*, 1990.
- Mann S and Picard RW, "The projective group model for multiframe resolution enhancement", *First IEEE International Conference on Image Processing (ICIP'94)*, (submitted) 1994a.
- Mann S and Picard RW, "The virtual bellows: A new perspective on the rigid planar patch", TR 260, MIT Media Lab Perceptual Computing Section, Cambridge, Ma. January 1994b.
- Mann S and Picard RW, "Virtual Bellows: constructing high quality stills from video", *IEEE International Conference on Image Processing*, pp. 363-367, 1994c.
- Mann S and Picard RW, "Video orbits of the projective group: A new perspective on image mosaicing", *MIT Media Laboratory*. Technical report No. 338 (Perceptual Computing Section), 1995.
- Mann S and Picard RW, "Video Orbits of the Projective Group: A New Perspective on Image Mosaicing", *Technical report, MIT*, 1996.
- Mann S, "Compositing multiple pictures of the same scene. Generalized large-displacement 8-parameter motion", *Proceedings of the 46th Annual IS&T Conference*. The Society of Imaging Science and Technology. Cambridge. Massachusetts, May 9-14 1993.
- McMillan L and Bishop G, "Plenoptic modelling: an image-based rendering system", *SIGGRAPH'95 Proceedings*, pp. 39-46, 1995.
- Meehan J, "Panoramic Photography", *Watson-Guption*, 1990.
- Milgram DL, "Computer Methods for Creating Photomosaics", *IEEE Transactions on Computers*, Vol. C-24, pp. 1113-1119, 1975.
- Milgram DL, "Adaptive Techniques for Photo Mosaicing", *IEEE Transactions on Computers*, Vol. C-26, pp. 1175-1180, 1977.
- Miller G et al., "The virtual museum: Interactive 3D navigation of a multimedia database", *Journal of Visualization and Computer Animation*, 3(3): 183-198, 1991.
- Millerson G, "TV Camera Operation", *Focal Press, London*, 1973.

- Mostafavi H and Smith FW, "Image correlation with geometric distortion Part I: Acquisition performance", *IEEE Trans. AES*, vol. AES-14, No. 3, pp. 487-493, 1978.
- Nayar SK, "Omnidirectional Vision", *Proceedings of the British Machine Vision Conference*, Vol. 1, pp. 1-12, 1998.
- Newton I, "Principia", Andrew Motte's translation, University of California Press. 1686.
- Nishihara K, "PRISM: a practical real-time stereo matcher", *MIT AI Memo 780*, 1984.
- Omnicam, *Columbia University*, <http://www.cs.columbia.edu/CAVE/omnicam>
- Peleg S and Herman J, "Panoramic mosaics by manifold projection", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 338-343, San Juan, Puerto Rico, June 1997a.
- Peleg S and Herman J, "Panoramic mosaics with VideoBrush", *IUW-97*, New Orleans, Louisiana, Morgan Kaufmann, May 1997b.
- Peleg S, "Elimination of seams in photomosaics", *Computer Graphics and Image Processing*, 16:90-94, May 1981.
- Porter T and Duff T, "Compositing digital images", *Computer Graphics (SIGGRAPH'84)*, 18(3): 253-259, July 1984.
- Press WH, Flannery BP, Teukolsky SA and Vetterling WT, "Numerical Recipes in C: The Art of Scientific Computing", *Cambridge University Press*, Cambridge, England, 1992.
- QuickTime VR, *Apple*, <http://www.apple.com/quicktime>
- RealVR, <http://www.rlspace.com>
- Rouso B, Peleg S and Finci I, "Generalized panoramic mosaics", *DARPA Image Understanding Workshop*, New Orleans, Louisiana, Morgan Kaufmann, May 1997a.
- Rouso B, Peleg S and Finci I, "Video Mosaicing using Manifold Projection", *Proceedings of the British Machine Vision Conference*, Vol. 1, pp. 1-10, 1997b.
- Sawhney HS and Kumar R, "True Multi-Image Alignment And Its Application To Mosaicing And Lens Distortion Correction", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 450-456, 1997.
- Sawhney HS, Ayer S and Gorkani M, "Model-based 2D & 3D dominant motion estimation for mosaicing and video representation", *Fifth International Conference on Computer Vision*, pages 583-590, Cambridge, MA, June 1995.
- Schalkoff RJ, "Development of High-Resolution Digital Image Mosaics from Low-Resolution Sensors Using Image Processing Techniques", *Proceedings of the Annual Southeastern Symposium on System Theory*, pp.42-46, 1986.

- Seitz S and Dyer C, "Physically valid view synthesis by image interpolation", *Proceedings IEEE Workshop on Representation of Visual Scenes*, Cambridge, MA, June 1995.
- Smooth Move, *Infinite Pictures*, <http://www.smoothwave.com>
- Spin Panorama, *Picture Works Technology*, <http://www.pictureworks.com/spinpano/index.html>
- Surround Video *Microsoft*, <http://www.bdiamon.com>
- Szeliski R and Coughlan J, "Hierarchical Spline-Based Image Registration", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pp. 194-201, Seattle, Washington, June 1994.
- Szeliski R and Heung-Yeung S, "Creating Full View Panoramic Mosaics and Environment Maps", *SIGGRAPH*, 1997.
- Szeliski R and Kang SB, "Direct Methods for Visual Scene Reconstruction", *ICCV Workshop on the Representation of Visual Scenes*, 1995.
- Szeliski R, "Image Mosaicing for Tele-Reality Applications", *Cambridge Research Lab Technical Report*, CRL 94/2, May 1994a.
- Szeliski R, "Video Mosaics for Virtual Environments", *IEEE Workshop on Applications of Computer Vision*, pp. 44-53, 1994b.
- Tekalp AM, Ozkan MK and Sezan MI, "High-Resolution Image Reconstruction from Lower-Resolution Image Sequence and Space Varying Image Restoration", *Proceedings of the International Conference on Acoustic, Speech and Signal Processing*, San Francisco, CA, Mar. 23-26, pp. III-169, 1992.
- Voronoi M G, "Nouvelles Applications des Parametres Continus a la Theorie des Formes Quadratiques", *J. Reine u. Angew. Math.*, 134:198-287, 1908.
- White G, "Video Techniques", *Heinnmann Professional Publishing*, London, 1982.
- Williams L, "Pyramidal Parametrics", *Computer Graphics (Proc. Siggraph)*, Vol. 17, No. 3, pp. 1-11, July 1983.
- Witkin A, Terzopoulos D and Kass M, "Signal matching through scale space", *International Journal of Computer Vision*, 1:133-144, 1987.
- Wolberg G, "Digital image warping", *IEEE Computer Society Press*, Los Alamitos, California, 1990.
- Wolf PR, "Elements of photogrammetry (with air photo interpretation and remote sensing)". - New York; London : McGraw-Hill, 1974.
- Zappala A, Gee A and Taylor M, "Document Mosaicing", *Proceedings of the British Machine Vision Conference*, Vol. 2, pp. 600-609, 1997.

Zoghiani I, Faugeras O and Deriche R, "Using Geometric Corners to Build a 2D Mosaic from a Set of Images", *Proceedings CVPR*, 1997.

Appendix A CD-ROM

The CD-ROM attached to this thesis contains the software developed for this research project. It also contains the original image sequences of the experiments described in sections 6.4 and 6.5, as well as the resulting mosaics. The VRML file with the spherical mosaic is included which can be viewed with an internet browser. It is recommended to use the VRML plug-in CosmoPlayer 2.0.

Contents of CD-ROM

DOCUMENT

BMVC98	Paper "Improved Mosaic Construction by Accumulated Alignment Error Distribution" <i>BMVC'98</i> (see Appendix B)
IPA99	Paper "Spherical Mosaic Construction using Physical Simulation for Consistent Image Alignment", <i>IPA'99</i> (see Appendix B)

EXPERIM

Data concerning the experiments carried out during the research

FULL_SPH	Full Spherical Mosaic Experiment (see section 6.5)
ALIGN	Registration data for all neighbour images
IMAGES	Image files (TGA)
MOSAICS	Resulting mosaics
VRML	VRML files as shown in Figure 6-27
LAB_PLAN	Partial Spherical Mosaic Mapped to a Plane (see section 6.3)
TEXT1	Document Mosaic (see section 5.6)
ALIGN	Registration data for all neighbour images
IMAGES	Image files (Zipped TGA)
MOSAICS	Resulting mosaics
TEXT2	Document Mosaic with grid (see section 5.6)
ALIGN	Registration data for all neighbour images
IMAGES	Image files (Zipped TGA)
MOSAICS	Resulting mosaics

WHITENDA	Double Band Panoramic Mosaic (see section 6.4)
ALIGN	Registration data for all neighbour images
IMAGES	Image files (TGA)
MOSAICS	Resulting mosaics
PROGRAMS	Source code of programs used during the research
ALIGN	This program calculates the transformation (Δx , Δy , $\Delta \theta$) that aligns two images Compiler: Borland C++ 3.1, MS-DOS Input: file "input.dta" containing the path to the TGA images and the requested pairs and the initial transformation to start the search Output: "output.dta" containing the computed transformations
FLICK	Tool used to verify the correct alignment of two images
EXAMPLE	Note: Make a copy in the hard disk to execute
P_SIMULA	
3D_ACCEL	Physical Simulation using hardware acceleration
DIALOG	Control Panel for Physical Simulation
GETNEIGH	Tool used by Physical Simulation to compute the neighbour images of a given image
NO_ACCEL	Physical Simulation without using hardware acceleration
SPH_MESH	Spherical Meshes
TEXTMAP	Texture Mapping
VORONOI	Program that demonstrates the Voronoi Tessellation
VRML	VRML files for the Full Spherical Mosaic as shown in Figure 6-27

Appendix B Publications

Guillén M, Holifield P and Varley M, “Improved Mosaic Construction by Accumulated Alignment Error Distribution”, *Proceedings of the British Machine Vision Conference*, Vol. 2, pp. 377-387, 1998.

Guillén M, Holifield P and Varley M, “Spherical Mosaic Construction using Physical Simulation for Consistent Image Alignment”, *Proceedings of the Seventh International Conference on Image Processing and its Applications IPA99*, Vol. 1, pp. 311-315, 1999.

Guillén M, Holifield P and Varley M, “Spherical Mosaic Construction using Physical Simulation for Consistent Image Alignment”, submitted for review for the IPA99 Special Section of the *IEE Proceedings on Vision, Image and Signal Processing*.

Improved Video Mosaic Construction by Accumulated Alignment Error Distribution

Manuel Guillén González Phil Holifield Martin Varley
Department of Engineering and Product Design
University of Central Lancashire
Preston PR1 2HE, UK
m.guillen-gonzalez@uclan.ac.uk

Abstract

Mosaic techniques have been used to obtain images with a large field of view from video sequences by assembling individual overlapping images. In existing methods of mosaic construction only consecutive frames are aligned. Accumulation of small alignment errors occur, and in the case of the image path returning to a previous position in the mosaic (looping path), a significant mismatch between non-consecutive frames will result. A new method for ensuring the consistency of the positions of all images in a mosaic is proposed. From the resulting improvement in mosaic quality, the new method enables construction of mosaics with a very large field of view.

1 Introduction

In recent years, computers have experienced a huge expansion of transmission, storage and processing capabilities, at the same time they have become commonplace in our homes. Video capture technology is available at low prices, but often does not give good resolution or field of view. Video mosaicing is a convenient way to capture images without such limitations.

Since the beginning of photography, mosaics have been used to obtain images with a larger field of view by assembling two or more individual overlapping images [1]. Today's applications include the scanning of large realistic images from the real world to generate virtual environments [2].

The construction of mosaics from video begins with the alignment of successive images. Using their relative positions, the images can be integrated in a single large picture. Projective transformation and lens distortion have been successfully modelled [3], so image alignment is not a major problem in mosaic construction. The aim of this paper is to show that a new step must be introduced in the mosaicing process to account for problems that occur when the camera follows a loop (looping path). This is the case, for example, when a camera pans in one direction then pans back to the starting position. In almost all existing methods of image registration, consecutive

frames of a video sequence are aligned. Accumulation of small alignment errors occur, and in the case of the image path returning to a previous position in the mosaic, a significant mismatch between non-consecutive frames will result.

It has been shown [4, 5] that instead of aligning successive images, the alignment can be done between an image and the actual mosaic as it is being composited. This may be an improvement with respect to the frame-to-frame alignment method, but loops involving large numbers of images result in distortions in the mosaic.

A new method for ensuring the consistency of the positions of all images in the mosaic is proposed, resulting in general improvement of mosaic quality and making it possible to create mosaics with a very large field of view, including spherical mosaics. The spherical mosaic is a two-dimensional mosaic mapped onto a sphere, and has applications in virtual reality environment maps [6].

2 Mosaic Construction

The basic processing steps involved in video mosaic construction are well known [7], and can be summarised as follows:

Image alignment: Determines the transformation that aligns two successive images, or one image with the current mosaic. In some cases simple translation and rotation operations correctly describe the transformation, in others, a projective transformation is needed.

Image integration: Consists of the selection of non-overlapping areas in the images that will contribute to the final mosaic or the combination of pixel intensities from overlapping images. Further blending of neighbour images is necessary to reduce the visibility of seams due to differences in intensities.

2.1 Image Alignment with Progressive Complexity

To determine the transformation that aligns two images, the different existing techniques can be divided in two types. The first identifies and matches common features in a pair of images such as lines [8], corners, text [9], etc. and uses them as references to align the two images. This method imposes limitations on the content of the images being aligned, for they must contain such features. The second type finds the transformation that minimizes the sum of the squared intensity errors for all overlapping pixels as shown in (1), therefore relying on the pixel intensities as features. For this method to work properly, there must be intensity variations in the images.

$$E = \sum_n^N [I_i(x_n, y_n) - I_j(x'_n, y'_n)]^2 \quad (1)$$

where I_i is image i
 I_j is image j after the transformation is applied
 N is the number of overlapping pixels
 x_n, y_n and x'_n, y'_n are related by the transformation matrix

A rigid transformation would only involve translation and rotation of the images, while a projective transformation requires more parameters to be considered. The distortion introduced in the acquisition process by the camera lens must also be modelled and corrected if accurate alignments are to be obtained.

The rigid transformation that aligns image I_i with image I_j is given using homogeneous coordinates in matrix form as follows.

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & d_x \\ \sin\theta & \cos\theta & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} \quad (2)$$

where d_x , d_y are the translation in pixels and θ is the angle of rotation.

It has been shown that in image alignment, the use of transformation models with progressive complexity reduces the computation cost [4, 10]. Although for the rigid transformation there are only three parameters to be computed, two for the translation and one for the angle of rotation, a simpler model involving translation solely can be used initially for the alignment. Then, using this translation component, the angle of rotation can be computed.

```
DO {
    shift = CALC_TRANSLATION(image i, image j)
    TRANSLATE_IMAGE(image j, shift)

    angle = CALC_ROTATION(image i, image j)
    ROTATE_IMAGE(image j, angle)
} UNTIL better accuracy cannot be achieved
```

Figure 1: Steps for calculating the translation and rotation alignment between two overlapping images using transformation models with progressive complexity.

This particular implementation starts by finding the translation (CALC_TRANSLATION) that best aligns the pair of images, that is, parameters d_x and d_y in (2). This offset does not account for the rotation between the images, but since the angle of rotation is small between successive images it may be neglected at this stage. Then, once the translation has been calculated, the angle θ is worked out (CALC_ROTATION). This process of adjusting translation and then rotation is repeated in several passes until better accuracy cannot be achieved. This normally requires no more than 2 or 3 passes. The steps are summarised in Figure 1.

Although the accuracy that can be achieved in image alignment is excellent when assessed by the human eye, it is not error-free. A small error is always present and will manifest itself after a number of successive image alignments.

2.2 Translation

Laplacian pyramids have been used for the computation of translations. Smaller images are created from the original images by averaging blocks of pixels. A translation is computed for these smaller images which is then used as an initial

position to compute the translation of the original images. A detailed description can be found in [10].

It has been found that it is important to follow all local minima to the next pyramid level, especially for document images where false matches at the lower resolution levels may occur due to the repetitive nature of text lines.

2.3 Rotation

Rotating an image is a time consuming operation, in particular when subpixel accuracy by means of interpolation is necessary. The process that has been used for the computation of the angle of rotation between two images involves the mapping of a circle onto a rectangle as shown in Figure 2. This warping takes a radial line from the source image (e.g. Figure 2a) and maps it into a row (e.g. Figure 2c), corresponding directly to a mapping from the polar coordinate system to the Cartesian coordinate system, i.e. $(r, \theta) \rightarrow (x, y)$.

A vertical translation in the warped image (Figure 2c) is equivalent to a rotation in the original image (Figure 2a). Thus the problem of minimising the error function for different angles is reduced to a vertical translation matching, which is computationally less expensive and uses the same algorithms developed for translation alignment.

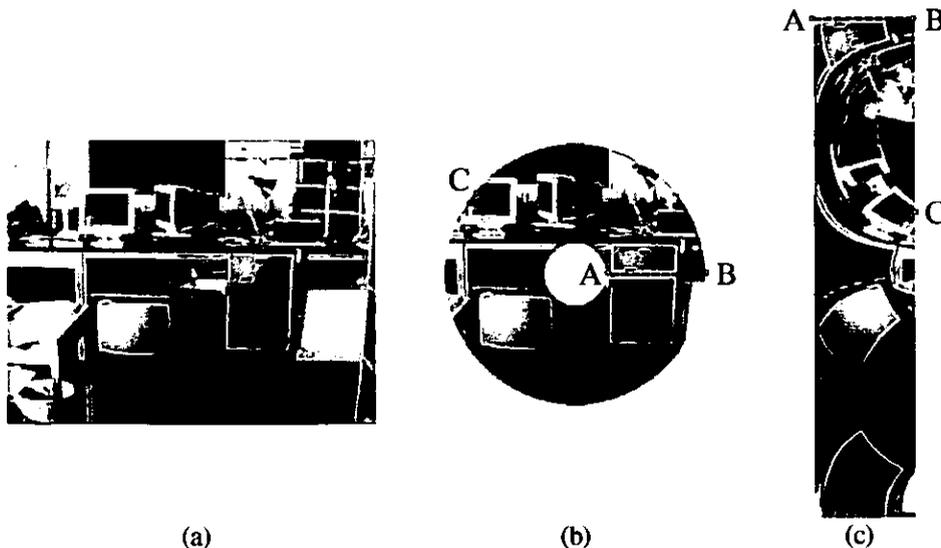


Figure 2: By warping from the polar to the Cartesian coordinate system, finding the alignment angle between two images is reduced to a vertical matching. (a) Original image. (b) Area to be warped onto a rectangle. (c) Effect of warping a circle onto a rectangle.

2.4 Image Integration

Once the position of the images are known they can be integrated in the mosaic. Each pixel in the mosaic is taken from the image whose centre is the nearest among all image centres. This scheme corresponds to Voronoi tessellation [11], which, given the

position of the centres of the images, defines a polygonal area for each image that will be pasted on the mosaic.

3 Looping Path Problem

In constructing mosaics from video sequences, almost all existing methods have used parameters computed by successive image alignment. Cumulative alignment errors occur when the position of images in the mosaic is based on successive image alignment only.

Although good alignment is achieved between successive images, cumulative errors cause poor alignment when the image path follows a loop, i.e. when the same area of the scene is covered by images which are distant in the sequence. This problem has only been identified in literature [4, 5], and has not been satisfactorily solved.

In Figure 3, assuming a perfect loop has been followed by the camera, images 1 and 60 should overlap perfectly, but misalignment error occurs due to accumulation of small errors in each successive image alignment. The effects of the *looping path problem* are dramatic when large numbers of images are involved in the loop. In addition, the misalignment of neighbour images is unavoidable even when the frame-to-frame displacement has been computed very accurately.

The alignment between images 1 and 60 in Figure 3 is inconsistent with the position of the rest of the images. Previous attempts at solving this inconsistency align the images with the mosaic as it is being composited [12]. Using this approach will result in a poor quality mosaic when a large number of images are involved in a loop, which is the case for large field of view mosaics. There will be cases where the next image to be aligned with the current mosaic will need to fit two or more different transformations and distortion will be inevitable (e.g. image 60 in Figure 3 will be distorted when aligned with image 59 and with image 1).

A new step must be introduced in the mosaicing process to account for the looping path problem. The proposed solution seeks to distribute the accumulated error of the positions of all images in the mosaic.

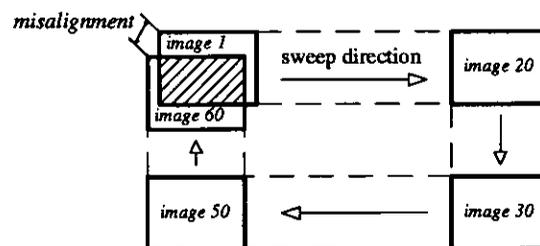


Figure 3: Misalignment error between image 1 and image 60 due to accumulation of small errors in successive image alignment.

3.1 Solution

Neighbour images are those which share a boundary in the mosaic. Each pair of neighbour images are related by a relative position t_{ij} computed using the alignment

method explained in section 3. For the rigid model, a translation (d_x, d_y) and a rotation angle d_θ of image j with respect to image i , correctly describes their relative position.

$$\mathbf{t}_{ij} = \begin{pmatrix} d_x \\ d_y \\ d_\theta \end{pmatrix} \quad (3)$$

A premise is introduced here: the relative position of a neighbour pair of images can be modified slightly without introducing a visible loss in quality. Such a change from its computed position must not exceed a fraction of a pixel if the seam is to remain unnoticeable.

\mathbf{T}_{ij} represents the correct relative position that aligns images i and j consistently along with all other images in the mosaic. The cost for this consistency is a slight modification Δ_{ij} of the computed relative positions of the images.

$$\mathbf{T}_{ij} = \mathbf{t}_{ij} + \Delta_{ij} \quad \Delta_{ij} = \begin{pmatrix} \delta_x \\ \delta_y \\ \delta_\theta \end{pmatrix} \quad (4)$$

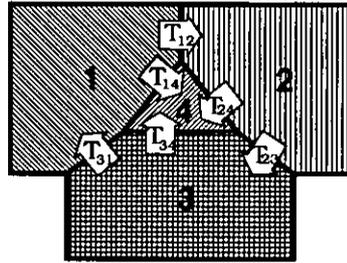


Figure 4: Example of mosaic composed from a sequence of 4 images.

In the sequence of 4 images shown in Figure 4, the pairs (1,2), (2,3), (3,4), (3,1), (1,4), (2,4) are neighbour images. The transformations that align them are \mathbf{t}_{12} , \mathbf{t}_{23} , \mathbf{t}_{34} , \mathbf{t}_{31} , \mathbf{t}_{14} and \mathbf{t}_{24} . An equation can be established for each possible route connecting the images:

$$\begin{aligned} \mathbf{T}_{12} \oplus \mathbf{T}_{23} &= \mathbf{T}_{13} & \therefore & (\mathbf{t}_{12} + \Delta_{12}) \oplus (\mathbf{t}_{23} + \Delta_{23}) = (\mathbf{t}_{13} + \Delta_{13}) \\ \mathbf{T}_{12} \oplus \mathbf{T}_{24} &= \mathbf{T}_{14} & \therefore & (\mathbf{t}_{12} + \Delta_{12}) \oplus (\mathbf{t}_{24} + \Delta_{24}) = (\mathbf{t}_{14} + \Delta_{14}) \\ \mathbf{T}_{23} \oplus \mathbf{T}_{34} &= \mathbf{T}_{24} & \therefore & (\mathbf{t}_{23} + \Delta_{23}) \oplus (\mathbf{t}_{34} + \Delta_{34}) = (\mathbf{t}_{24} + \Delta_{24}) \\ \mathbf{T}_{31} \oplus \mathbf{T}_{14} &= \mathbf{T}_{34} & \therefore & (\mathbf{t}_{31} + \Delta_{31}) \oplus (\mathbf{t}_{14} + \Delta_{14}) = (\mathbf{t}_{34} + \Delta_{34}) \end{aligned} \quad (5)$$

where \oplus means composition of transformations.

The minimum values of Δ_{ij} that satisfy equations 5 give the set of relative positions \mathbf{T}_{ij} that consistently align all images in the mosaic.

Although a solution can be found that minimises Δ_{ij} , its implementation becomes impractical for a large number of images, which is the case in reality. A different approach to the problem is therefore necessary.

3.2 Proposed algorithm

The proposed method for consistently aligning all images in the mosaic is explained in this section.

For a sequence of N images, the relative positions $\{t_{01}, t_{12}, t_{23}, \dots, t_{N-2 N-1}\}$ that align successive images are computed. Then, the initial positions of the images in the mosaic $\{P_0, P_1, P_2, \dots, P_{N-1}\}$ can be calculated, as shown in equation 6, by composition of the transformations that align successive images in the sequence.

$$P_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad P_i = P_{i-1} \oplus t_{i-1 i} \quad 1 \leq i \leq N-1 \quad (6)$$

where the meaning of \oplus is given in (7).

$$P_i \oplus t_{ij} = \begin{pmatrix} P_{x_i} \\ P_{y_i} \\ P_{\theta_i} \end{pmatrix} + \begin{pmatrix} t'_{x_{ij}} \\ t'_{y_{ij}} \\ t_{\theta_{ij}} \end{pmatrix} \quad \text{where} \quad \begin{pmatrix} t'_{x_{ij}} \\ t'_{y_{ij}} \end{pmatrix} = \begin{pmatrix} \cos(P_{\theta_i}) & -\sin(P_{\theta_i}) \\ \sin(P_{\theta_i}) & \cos(P_{\theta_i}) \end{pmatrix} \begin{pmatrix} t_{x_{ij}} \\ t_{y_{ij}} \end{pmatrix} \quad (7)$$

So far this corresponds to successive image alignment.

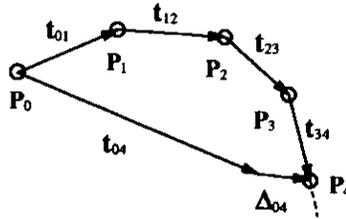


Figure 5: Relation between the positions of the images in the mosaic P_0, P_1, P_2, P_3, P_4 and the relative positions $t_{01}, t_{12}, t_{23}, t_{34}$ that align successive images. Δ_{04} is the error between the computed relative position of images 0 and 4 (t_{04}) and their actual relative position in the mosaic ($P_4 - P_0$). The circles represent the centres of the images.

Let Δ_{ij} be the difference between the constant relative position of images i and j (t_{ij}) and their relative position in the mosaic ($P_j - P_i$), which will be modified. Δ_{ij} represents the error between the computed relative position of images i and j and their actual relative position in the mosaic.

$$\Delta_{ij} = t_{ij} - (P_j - P_i) \quad (8)$$

Initially, for successive images (i.e. $j = i+1$), $\Delta_{ij} = 0$. For the rest of the neighbour images $\Delta_{ij} \neq 0$ due to the accumulated error in the successive alignments between the images i and j , that is, the error to be reduced. An analogy with a physical model is introduced, consisting of a network of connected nodes representing the centres of the images on which forces are exerted in order to change their position. The links between nodes are defined by the transformations that align neighbour images (see Figure 5). In this analogy $f(\Delta_{ij})$ represents the force pushing image i towards the right position with respect to image j , where the function f will be defined later in this section.

Let $\{n_1, n_2, n_3, \dots, n_m\}$ be the m neighbour images of image i . Each of its neighbour images will exert a force upon image i , Δ_i is the resultant summation of these forces.

$$\Delta_i = \sum_{k=1}^m \Delta_{i n_k} \quad (9)$$

The refinement process that leads to a consistent set of positions is accomplished in an iterative fashion. For each iteration the forces acting on all images are calculated, then their positions are modified accordingly. The loop ends when equilibrium is achieved, i.e. $\Delta_i \approx 0$, $0 \leq i \leq N-1$.

The positions of the images (P_i) are modified by small increments. These increments are a function of Δ_i .

$$P_i \rightarrow P_i + f(\Delta_i) \quad (10)$$

The performance of the function $f(\Delta_i)$ is assessed by inspection of the overall distortion E and the error for the worst case E_{\max} once the equilibrium is achieved.

$$E = \sum \text{abs}(\Delta_{ij}) \quad E_{\max} = \max\{\text{abs}(\Delta_{ij}), \forall i, j \text{ neighbour images}\} \quad (11)$$

Different approaches have been tried to model $f(\Delta_i)$. The function that gives the minimum error and the fastest convergence was found to be proportional to the square of Δ_i (shown in equation 12). The constant k is a small number required to maintain the stability of the system, since large increments lead recursively to even larger increments.

$$f(\Delta_i) = k \times \begin{pmatrix} \text{sign}(\delta_x) \times \delta_x^2 \\ \text{sign}(\delta_y) \times \delta_y^2 \\ \text{sign}(\delta_\theta) \times \delta_\theta^2 \end{pmatrix} \quad (12)$$

After the positions $\{P_1, P_2, P_3, \dots, P_N\}$ have been readjusted, new pairs of images may have become neighbours and some may no longer have common boundaries. In the case of new neighbour images appearing, their alignment transformation will be computed and the readjustment process performed again until no new neighbours arise.

At the end of the process the cumulative error is spread across all images, and therefore no single pair of images show a marked misalignment.

4 Experimental Results

The technique for solving the looping path problem has been tested with various sets of images resulting in excellent overall improvement of the mosaic. See Figures 6 and 7 for illustrative results.

The errors in the positions of the images (E and E_{\max} in equation 11) are shown in Tables 1,2 and 3, for three different mosaics. 'Text 1' is a mosaic of a text document. Mosaic 'Text 2' is the same text with a superimposed grid used to assess visually the quality of the seams. The images for the mosaic labelled 'Lab' were obtained with a hand held video camera from a fixed location, and despite the parallax and the distortion from mapping of a spherical view on a flat image, the results are promising (Figure 8).

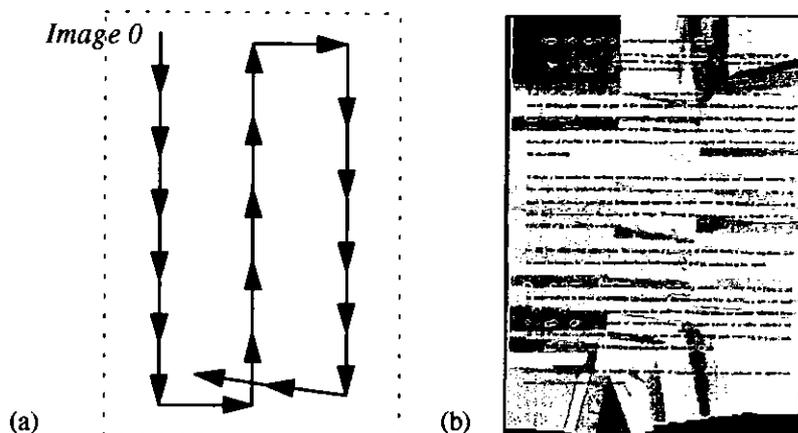


Figure 6: (a) Path followed by the camera. (b) Mosaic 'Text 1' (1663x2320 pixels), composed of 141 images (736x560 pixels). The grey levels represent the areas in the mosaic used from each particular image.



Figure 7: Details of mosaic 'Text 1'. Left, successive image alignment only, showing looping path problem. Right, corrected positions of images.

'Text 1'	E_{\max}	Between Images	Sum Error E	Average Error per seam
x (pixels)	0.6823	128 and 118	63.11	0.1783
y (pixels)	0.5386	105 and 57	75.63	0.2136
θ (rads)	0.002597	7 and 6	0.3728	0.001053

Table 1: Mosaic 'Text 1', 141 images, 354 pairs of neighbour images. The table shows the errors present after readjustment. In the worst case the images have been displaced by about half a pixel from their computed position.

'Text 2'	E_{\max}	Between Images	Sum Error E	Average Error per seam
x (pixels)	0.5627	71 and 65	28.56	0.1527
y (pixels)	0.5162	16 and 15	46.35	0.2479
θ (rads)	0.003333	71 and 65	0.1636	0.0008749

Table 2: Mosaic 'Text 2', size 1481x2139 pixels, 80 images, 187 pairs of neighbour images. Final errors.



Figure 8: Mosaic 'Lab'. Left, successive image alignment only, showing looping path problem. Right, corrected positions of images.

'Lab'	E_{\max}	Between Images	Sum Error E	Average Error per seam
x (pixels)	0.9934	123 and 62	71.64	0.2372
y (pixels)	1.0576	82 and 81	77.65	0.2571
θ (rads)	0.01610	123 and 65	1.186	0.003927

Table 3: Mosaic 'Lab', 130 images, 302 pairs of neighbour images. The errors are higher than in the other mosaics due to errors in the computation of image alignment caused by parallax. In addition, the field of view is about 90° , so further distortions are introduced by the mapping onto a plane.

5 Conclusions

It has been shown that a new step must be introduced in video mosaic construction to account for the looping path problem. Cumulative errors occur in successive image alignment, and in the case of the image path returning to a previous position in the mosaic, a significant mismatch between non-successive images will result. The proposed solution makes use of the alignments between all neighbour images to consistently position the images on the mosaic. Starting with the successive image alignment positions, these are modified by small increments to reduce the overall misalignment error.

Since the field of view is not a limitation when using this approach, 360° mosaics can now be produced. Current research aims at full spherical mosaics for applications in Virtual Reality.

Considering the projective transformation matrix as a representation of the position and orientation of a camera in space, an analogous method using forces can be used for the consistent alignment of all images in the mosaic using the projective model.

6 References

- [1] Paul R. Wolf, "Elements of photogrammetry (with air photo interpretation and remote sensing)". - New York; London : McGraw-Hill, 1974.
- [2] R. Szeliski, "Video Mosaics for Virtual Environments", IEEE Workshop on Applications of Computer Vision, pp. 44-53, 1994.

- [3] S. Mann and R. W. Picard, "Virtual Bellows: constructing high quality stills from video", IEEE International Conference on Image Processing, pp. 363-367, 1994.
- [4] H. S. Sawhney, R. Kumar, "True Multi-Image Alignment And Its Application To Mosaicing And Lens Distortion Correction", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 450-456, 1997.
- [5] P. J. Burt, M. Hansen, P. Anandan, "Video Mosaic Displays", Proceedings of SPIE - International Society for Optical Engineering, Vol. 2736, pp.119-127, 1996.
- [6] R. Szeliski, "Image Mosaicing for Tele-Reality Applications", Cambridge Research Laboratory, Technical Report Series, May 1994.
- [7] S. Peleg and J. Herman, "Panoramic mosaics by manifold projection", IEEE Conference on Computer Vision and Pattern Recognition, pp. 338-343, San Juan, Puerto Rico, June 1997.
- [8] Rik D. T. Janssen and A. M. Vossepoel, "Compilation Of Mosaics From Separately Scanned Line Drawings", IEEE Workshop on Applications of Computer Vision - Proceedings, pp. 36-43, 1994.
- [9] A. Zappala, A. Gee, M. Taylor, "Document Mosaicing", Proceedings of the British Machine Vision Conference, Vol. 2, pp. 600-609, 1997.
- [10] M. Hansen, P. Anandan, K. Dana, G. van der Wal, P. Burt, "Real-time Scene Stabilization and Mosaic Construction", IEEE Workshop on Applications of Computer Vision - Proceedings, pp.54-62, 1994.
- [11] F. Aurenhammer, "Voronoi diagrams: A survey of a fundamental geometric data structure", ACM Comp. Surv., 23:345-405, 1991.
- [12] M. Irani, P. Anandan, S. Hsu, "Mosaic Based Representations of Video Sequences and Their Applications", IEEE International Conference on Computer Vision, pp. 605-611, 1995.

SPHERICAL MOSAIC CONSTRUCTION USING PHYSICAL ANALOGY FOR CONSISTENT IMAGE ALIGNMENT

M. Guillén González, P. Holifield, M. R. Varley

University of Central Lancashire, Preston, U.K.

ABSTRACT

Mosaics are images with a large field of view obtained by assembling individual overlapping images. Two images recorded by a rotating ideal camera are related by the same transformation that relates the camera's sensor plane positions and orientations in space at the time the images were captured. Through the intersection line of the sensor planes, the intensity values of both images coincide. This intersection line can be extracted by image registration. A method to construct spherical mosaics is presented which hinges overlapping images to find a consistent position that solves the problem of accumulation of errors in successive image alignment (looping path problem). A physical simulation, offering interactivity, modularity and visualisation capabilities, is used to hinge the images through a set of forces modelled to converge to a stable solution. Images are correctly placed on the surface of a sphere of initially unknown radius (i.e. without knowing the focal length of the camera).

INTRODUCTION

Mosaics are images with a large field of view obtained by assembling individual overlapping images. Mosaics can be created from video sequences recorded by a variety of camera motions, and in each case will produce flat mosaics, panoramic mosaics, or spherical mosaics. Panoramic mosaics offer the viewer the freedom to choose the direction of view as if virtually placed at the centre of a cylinder on the interior of which the mosaic is pasted. The user can choose to look left and right, zoom in and out, and, restricted to the resolution of the camera, up and down [1]. A natural extension of a panoramic mosaic is the spherical mosaic. The spherical mosaic is mapped onto the interior surface of a sphere, allowing views in all directions, including up and down.

In mosaic construction, accumulation of small alignment errors occurs when only consecutive frames are aligned, and in the case of the image path returning to a previous position in the mosaic a significant mismatch between non-consecutive frames will result [2], known as the looping path problem.

A solution that ensures the consistency of the positions of all the images for the spherical mosaic is presented. The solution minimises the error of the relative positions of all neighbour images that are integrated in the mosaic. This minimisation problem is implemented using an analogy with a physical model. In this model, each image is represented by a rigid body with unit mass, and forces are exerted between images to attract them to the correct position with respect to each other. Using this analogy, auxiliary forces can be introduced logically into the model to guide the images to the stable equilibrium state and to introduce the constraints needed for the spherical distribution of images. The simulation also provides means of assessing the convergence by inspection of the energy of the system.

IMAGE ALIGNMENT

The motion of the camera determines the type of mosaic that can be constructed. For a camera free to pan about its optical axis, the images captured are related by a horizontal translation and the resulting mosaic is a cylinder (Fig. 1a). For a camera fixed over a desk, as a document is passed under the camera, different portions of it become visible. Any two frames are related by a translation and a rotation, corresponding to a rigid transformation (Fig. 1b). The transformation between images captured by a camera undergoing purely rotations about its optical axis (Fig. 1c) can be approximated to the rigid transformation if the rotation of the camera between frames is a small angle [3]. The images are related by a rotation and a translation in two dimensions mapped on the interior surface of a sphere.

The rigid transformation, given in matrix form (see [1]), using homogeneous coordinates, is as follows

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & \Delta_x \\ \sin \theta & \cos \theta & \Delta_y \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ w \end{bmatrix} \quad \text{Eq. 1}$$

where Δ_x , Δ_y are the horizontal and vertical translations in pixels and θ is the angle of rotation (see Fig. 2).

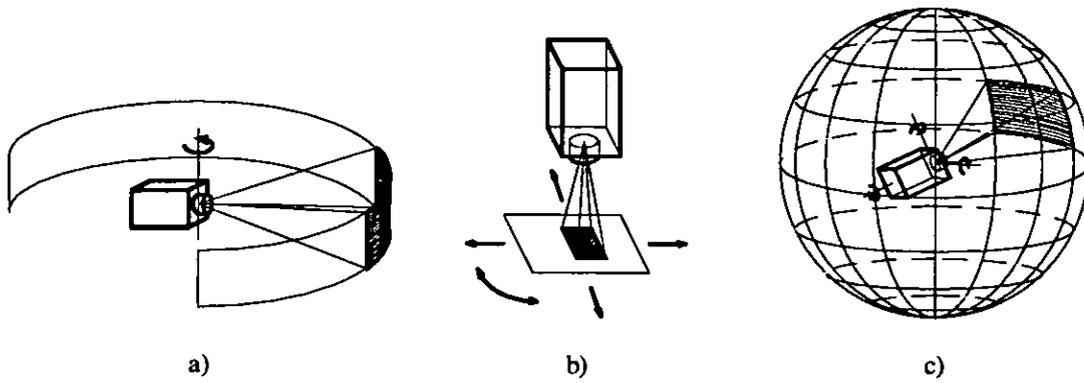


Fig. 1 a) Cylindrical mosaic. b) Flat mosaic. c) Spherical mosaic.

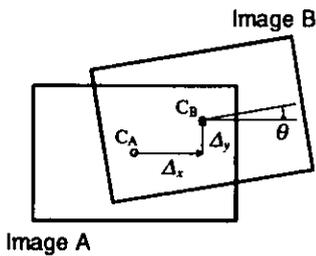


Fig. 2 Rigid transformation that aligns two overlapping images. C_A and C_B are the centres of images A and B respectively.

The alignment between two images can be computed by identifying common features in both images (lines, corners, text, etc.) and extracting their relative position, or by finding the transformation that minimises the sum of the squared intensity errors for all overlapping pixels [1]. Regardless of the method used or the complexity of the transformation matrix, a small error will always be present in each alignment and it will become apparent when accumulated through a number of images (looping path problem).

SOLUTION TO THE LOOPING PATH PROBLEM

A solution to the looping path problem has been proposed [4] which iteratively refines the positions of the images on the mosaic. This method starts by computing the alignment of successive images to obtain an initial position of the images on the mosaic. Images which are new neighbours (i.e. those which now share a boundary in the mosaic) are then identified. The alignment of the new neighbours is computed and the error function is minimised for all pairs of neighbour images by modifying their positions. This is repeated until no new neighbours arise.

This process can be summarised as shown in Fig. 3.

```

Compute alignment of successive images.
Identify neighbour images.

WHILE (new neighbours arise) {
  1. Compute alignment of new neighbours.
  2. Minimise error function by changing
     positions of images.
  3. Identify new neighbour images.
}
  
```

Fig. 3 Pseudocode for solving the looping path problem.

The error function to be minimised is the overall error in a mosaic, and is discussed in the next section.

PHYSICAL MODEL FOR THE IMAGES

Step 2 in Fig. 3 is a minimisation problem in which the cost function depends on the positions of the images in the mosaic and the relative computed alignments. An analogy with a physical model has been chosen to minimise the cost function for a number of reasons, which include visualisation, interactivity and modularity. This physical model must be devised for each image so that the images are attracted towards the computed relative positions. When the system reaches the rest state the cost function is minimised. A physical simulation permits the visualisation of the model. The topology of the set of images as they move towards equilibrium can be monitored and provides important feedback on the suitability of the model used. Although the evolution of the positions of the images is simple in the case of a planar mosaic, it is rather polymorphic in the case of the spherical mosaic in three-dimensional space. The modularity allows modelling of complex relationships using simple elements such as springs and dampers. The interaction with the model in the form of adding and removing constraints such as the constraints for the spherical mosaic is necessary in order to gain a comprehensive understanding of the effect of such constraints. A physical simulation has been chosen for this application for the reasons mentioned. However, other traditional minimisation algorithms such as Levenberg-Marquardt, have been used to solve the looping path problem [5].

The constraint used to impose the relative positions in a pair of overlapping images is a non-rigid hinge (see Figs. 4 and 5). A hinge is defined, for every pair of overlapping images, by two points in one image (P_{H1A} and P_{H2A}) which are attracted to the corresponding two points in the other image (P_{H1B} and P_{H2B}). This attraction is a force whose magnitude depends on the distance between corresponding points.

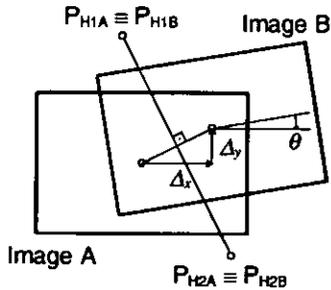


Fig. 4 The hinge model. When the two images are correctly aligned, the hinge points coincide ($P_{H1A} \equiv P_{H1B}$ and $P_{H2A} \equiv P_{H2B}$).

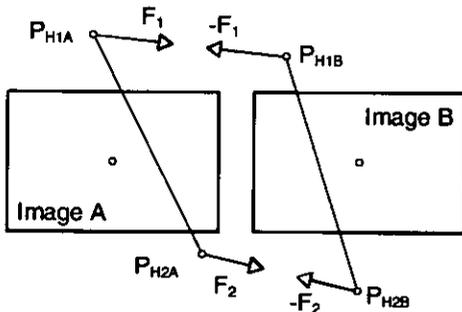


Fig. 5 When the images are separated from their relative positions, forces F_1 and F_2 will pull the images to correct positions.

The error function to be minimised can now be defined for N hinges as:

$$E = \sum_{i=1}^N (|P_{H1A_i} - P_{H1B_i}| + |P_{H2A_i} - P_{H2B_i}|) \quad \text{Eq. 2}$$

Figures 6 to 9 show the progression of a pair of overlapping images initially placed at a distance from each other.

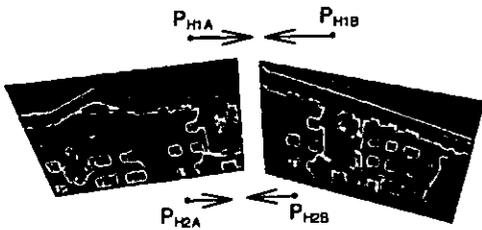


Fig. 6 Initial position. The arrows represent the forces pulling the hinge points together.

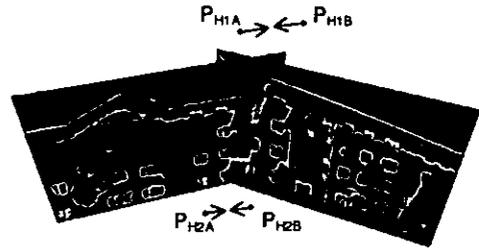


Fig. 7 After a few iterations the hinge points are closer to each other. There are still forces pulling the hinge points together.

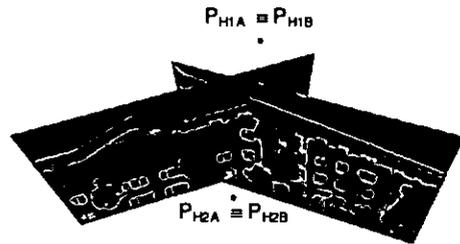


Fig. 8 At the end position, the hinge points overlap perfectly. In this position, the pixel intensities at the intersection are common to both images.



Fig. 9 The aligned images of Fig. 8 as seen from a view point directly in front of both images.

Good results have been obtained with forces proportional to the square of the distance between hinge points.

The Spherical Constraint

The second constraint investigated in this paper is the spherical constraint which makes the images face the same point, which, for convenience, is chosen to be the centre of mass (CM) of the system. To avoid a misplaced CM due to uneven distribution of images, the CM of each individual image can be situated at a distance in front of it, rather than at its centre. The spherical constraint imposes the images to be located

on the surface of a sphere, although its radius is not known *a priori*. This constraint is implemented with two forces of the same magnitude and opposite direction (see Fig. 10). The points of action of the forces are P_{CM1} and P_{CM2} , which are situated at a fixed distance in front and behind the centre of the image respectively.

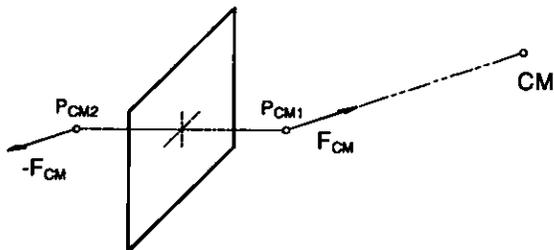


Fig. 10: Spherical constraint. Auxiliary forces acting on P_{CM1} and P_{CM2} rotate the image towards the centre of mass of the system.

Figs. 11 and 12 show a mosaic obtained from 189 images recorded with a hand-held video camera by panning 360° twice at different tilt angles.

CONCLUSIONS

Two images recorded by a rotating ideal camera are related by the same transformation that relates the camera's sensor plane positions and orientations in space at the time the images were captured. The intersection of the sensor planes is a line, and, through that line, the intensity values for both images coincide. This intersection line can be extracted by image registration using the rigid model if the rotation angle of the camera is small [3] [6].

When the position in space of two images is such that they cross through the intersection line, no misalignment is shown. In addition, if the images are viewed from the intersection point of the perpendiculars to the images passing through their respective centres, there will be no projective distortion [7]. However, the angle between the image planes depends on the field of view of the camera (or equivalently, its focal length), and needs to be derived.

A method to construct spherical mosaics has been presented which hinges overlapping images through their intersection line. It finds a consistent position of a set of images that solves the looping path problem. A physical simulation, offering interactivity, modularity and visualisation capabilities, is used to hinge the images through a set of forces which converge to a stable solution. By using the spherical constraint, and having a sufficient number of images, these are correctly placed on a sphere of initially unknown radius (i.e. without knowing the focal length of the camera).

Using inexpensive video cameras, high resolution mosaics can be produced by composition of video frames. In particular, spherical mosaics can be used as environment maps for games and other virtual reality applications. Spherical mosaics can be created as VRML files to be viewed using an internet browser with a VRML plug-in.

REFERENCES

1. Szeliski R., 1994, "Video Mosaics for Virtual Environments", IEEE Workshop on Applications of Computer Vision, 44-53.
2. Burt P., Hansen M. and Anandan P., 1996, "Video Mosaic Displays", Proceedings of SPIE - International Society for Optical Engineering, Vol. 2736, 119-127.
3. Mann S. and Picard R.W., 1994, "Virtual Bellows: constructing high quality stills from video", IEEE International Conference on Image Processing, 363-367.
4. Guillén-González M., Holifield P. and Varley M., 1998, "Improved Mosaic Construction by Accumulated Alignment Error Distribution", British Machine Vision Conference 1998 Proceedings Vol. 2, 377-387.
5. Capel D. and Zisserman A., 1998, "Automated Mosaicing with Super-resolution Zoom", Proceedings of the Conference on Computer Vision and Pattern Recognition, Santa Barbara.
6. Mostafavi H. and Smith F.W., 1978, "Image correlation with geometric distortion Part I: Acquisition performance", IEEE Trans. AES, vol. AES-14, No. 3, 487-493.
7. Gümüstekin S. and Hall R.W., 1996, "Mosaic Image Generation on a Flattened Gaussian Sphere", Third IEEE Workshop on Applications on Computer Vision, Florida, 50-55.

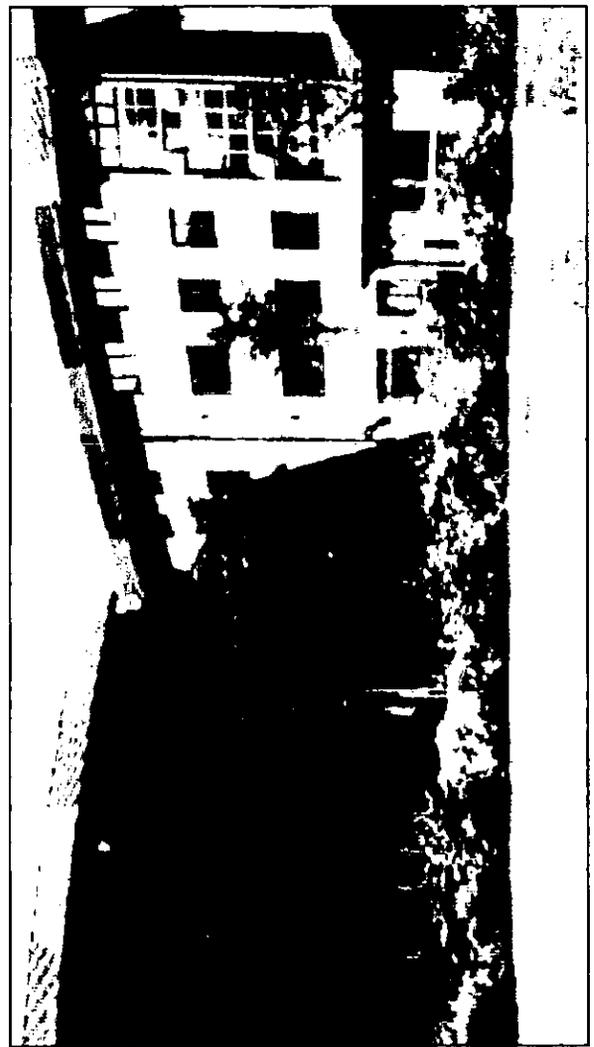
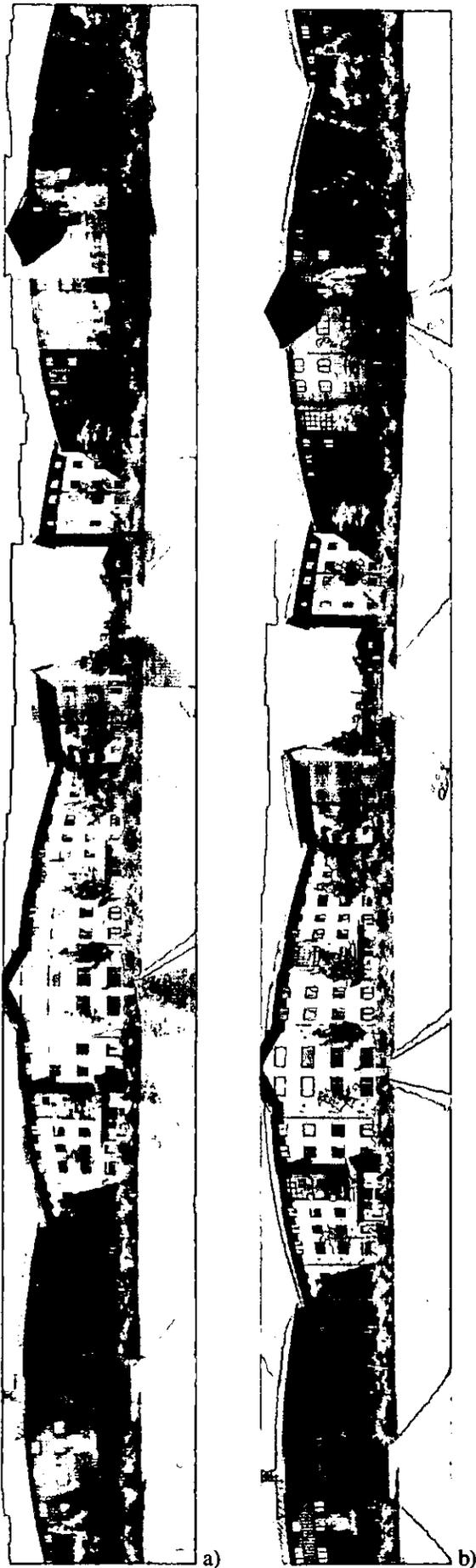


Fig. 12 Detail from the mosaic in Fig. 11a.

Fig. 11 Mosaic obtained from 189 images recorded with a hand-held video camera by panning 360° at different tilt angles. The coloured areas in a) represent the areas in the mosaic used from each individual image.

Spherical Mosaic Construction using Physical Analogy for Consistent Image Alignment

Manuel Guillén González, Phil Holifield and Martin R. Varley

ABSTRACT

Mosaics are images with a large field of view obtained by assembling two or more individual overlapping images. Two images recorded by a rotating ideal camera are related by the same transformation that relates the camera's sensor plane positions and orientations in space at the time the images were captured. The intersection line of the sensor planes can be extracted by image registration. A method to construct spherical mosaics is presented which connects overlapping images and finds consistent positions for the images, thus solving the problem of accumulation of errors in successive image alignment. A physical simulation, offering interactivity, modularity and visualisation capabilities, is used to minimise the alignment error by adjusting the positions of the images in three-dimensional space. Forces and constraints are applied such that the model converges to a stable solution. Experimental results are presented to demonstrate the construction of a spherical mosaic.

1 INTRODUCTION

Historically, the most significant application for image mosaicing is in the development of aerial images [1] and satellite images [2]. These mosaics are commonly used as maps and for remote sensing applications. With the arrival of digital photography, new applications for mosaicing were created [3-5]. The list of applications using mosaic techniques is extensive today. Mosaics can be useful where the image sensor does not have sufficient pixel resolution

to provide a single image with the level of detail needed. The attainable resolution in a mosaic is essentially unlimited [6].

Constructing mosaics from a number of overlapping images is a problem of image registration. Existing registration algorithms are designed to align pairs of images, but cannot be used with more than two images simultaneously. Since error-free image registration techniques do not exist, a new technique [7] for consistently aligning all images which form part of a mosaic is used, based on existing imperfect registration methods. In achieving this, the spherical mosaic, composed from images obtained by panning and tilting a camera and representing an omnidirectional view of the scene, is made possible. Experimental results demonstrating a spherical mosaic are presented.

2 THE LOOPING PATH PROBLEM AND PROPOSED SOLUTION

In video sequences, successive frames are guaranteed to overlap and, therefore, in mosaicing from video sequences, the registration is computed for successive images. A small alignment error, not normally visible, will exist regardless of the image registration method used and the accuracy achieved. However, accumulation of alignment errors occurs when consecutive images are aligned, and in the case of the image path returning to a previous position in the mosaic a significant visible mismatch between non-consecutive images will result: referred to as the *Looping Path Problem* (LPP). A proposed solution [7] to this which achieves consistent image alignment is summarised in Fig. 1.

```

1.   Compute alignments of successive images
2.   Create list of neighbour images
3.   WHILE (new neighbours arise) {
4.       Compute alignments of new neighbour images
5.       Minimise error by modifying positions of images
6.       Update list of neighbour images
    }

```

Figure 1 Pseudocode of process for consistent image alignment

The process begins by computing the alignment of successive images to obtain the initial positions of the images on the mosaic. Although the alignment error is accumulated at this stage, these positions are sufficiently correct to reveal new neighbour images (neighbour images are those that share a boundary in the mosaic). The list of neighbour images is updated according to the Delaunay triangulation [8].

A cost function is defined as the summation of the positional errors between all pairs of neighbour images. The positional error is a measure of the difference between the computed alignment and the actual position on the mosaic. The relative positions of neighbour images on the mosaic can be modified slightly without introducing a visible loss in quality. Such a change from the computed registration must not exceed a fraction of a pixel if the seam is to remain unnoticeable. Therefore, the cost function can be minimised by modifying the positions of the images on the mosaic, thereby reducing the gross visible error at the seams in a looping path. After the positions of the images have been rearranged to minimise the error, new neighbours arising are added to the list, and pairs of images which no longer share a boundary are deleted from the list. The list of neighbour images is updated, and the process continues until no new neighbours appear.

3 PHYSICAL ANALOGY FOR SPHERICAL MOSAIC CONSTRUCTION

3.1 Physical Model

Line 5 in Fig. 1 is a minimisation problem in which the cost function depends on the positions of the images on the mosaic and their relative computed alignments. This minimisation is achieved using an analogy with a physical model, in which each image is represented by a rigid body and forces are exerted to attract them to the correct relative positions. A system with any number of images can be simulated, and the evolution of the positions of the images towards the desired consistent positions may be monitored for optimisation.

An analogy with a physical model has been chosen for the following reasons:

- A physical model permits the *visualisation* of a set of images with different positions and orientations in space. The topology of the set of images as they move towards equilibrium can be monitored and provides important feedback on the suitability of the model used.
- The model is *modular*, allowing modelling of complex relationships by using simple elements such as springs and dampers.
- The model is *interactive*, allowing auxiliary forces and constraints required for particular cases, such as a spherical distribution of images, to be introduced in order to gain a comprehensive understanding of their effect on convergence and stability.

The model also provides a means of assessing the convergence to equilibrium by examining its energy. When the system reaches the rest state, its energy (that is, the sum of kinetic and potential energy) is minimum and its kinetic energy is zero. If different equilibrium states are reached, they can be compared in terms of the remaining potential energy, which indicates the

overall degree of alignment achieved in each case. Hence, the physical model helps to identify the forces necessary for the consistent alignment of images.

The model is simulated by computing the equations of motion of a physical system, consisting of one or more bodies, according to the forces exerted upon them. The variables typically investigated in a physical simulation include acceleration, velocity, and the spatial variables, i.e. position and orientation. These are given initial values, then the simulation is run. Variable values can be monitored during a simulation, and they are of great value in system design.

Each image is modelled as a rigid body with unitary mass and unitary moment of inertia, the centre of mass being situated at the centre of the image (see Fig. 2).

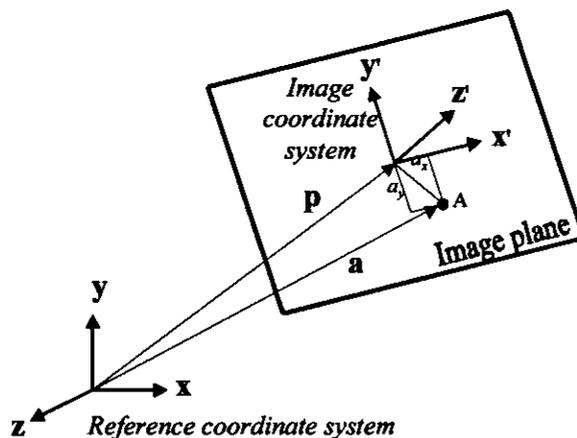


Figure 2 Position and orientation of an image as a rigid body

The position of an image in space is described as a vector \mathbf{p} , which is the translation of its centre from the reference coordinate system's origin.

The three unitary orthogonal vectors that define the image coordinate system are \mathbf{x}' , \mathbf{y}' and \mathbf{z}' . The vector axis \mathbf{z}' is perpendicular to the image plane, and \mathbf{x}' and \mathbf{y}' lie on the image plane and define the conventional horizontal and vertical axes used to locate the pixels on an image. A vector is said to be given in *image space* coordinates if its components are referred to the image coordinate system.

The position of a point A which lies on the image plane at image coordinates $\mathbf{a}_0 = (a_x, a_y, 0)$ is expressed in world coordinates as

$$\mathbf{a} = \mathbf{a}_0 \mathbf{S}_{\text{image}} + \mathbf{p} = \mathbf{p} + a_x \mathbf{x}' + a_y \mathbf{y}' \quad (1)$$

Two images recorded by a rotating ideal camera are related by a projective transformation. This is actually the transformation that relates the camera's sensor plane positions and orientations in space at the time the images were captured. Figure 3 shows the images projected onto the sensor plane of a camera as the camera rotates about its optical centre. The position and orientation of the sensor plane at the time the images were captured are closely related to the position and orientation of the images. The intersection of the sensor planes at these two different times is the *intersection line*, through which the intensity values for both images coincide. If the aligned images are viewed from the intersection point of the perpendiculars to the images passing through their respective centres, which corresponds to the optical centre, there will be no projective distortion [9].

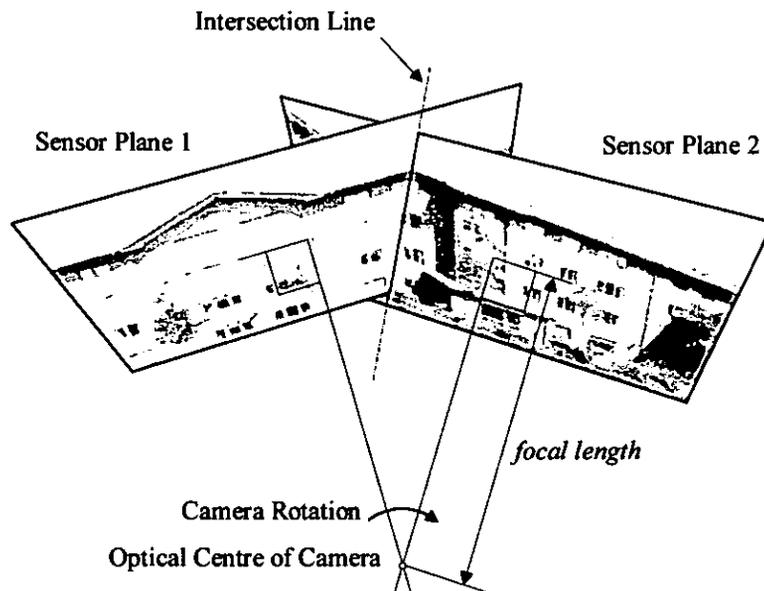


Figure 3 Two successive images projected onto the sensor plane of a camera

The intersection line can be extracted by image registration. If the rotation angle of the camera is small the rigid model can be used without introducing significant distortion [10]. Two images linked together behave as two masses connected by two springs. The role of the springs is to attract the images to their relative correct positions. Since we are not interested in the trajectory of the images but in their final equilibrium positions, the value of the spring constant k is not significant as long as equilibrium is eventually reached. Critically damped motion would be desirable for this application, since it provides the fastest convergence to the equilibrium state, therefore appropriate values must be chosen for the spring constant and the damping constant.

A damping agent is necessary to prevent unlimited oscillations, and can be thought of as a viscous fluid in which the inter-connected images are submerged, the damping forces exerted by the fluid being proportional to the velocity of the bodies moving in it. The angular velocity of the images must be damped as well, hence a suitable drag torque proportional to

the angular velocity should be present in the model to prevent oscillatory rotations of the images.

Initially, the images are positioned sequentially in a plane. The system has a potential energy stored in the elongated springs and no kinetic energy since it is at rest. When the simulation is run, the images start moving, the kinetic energy increases and the potential energy decreases accordingly. Some of the energy is dissipated due to the damping which subtracts energy from the moving images. It should be noted that friction only exists between the images and the fluid, and there is no friction between the images. When visualising the image system, it may seem that the images could end up entangled in a mess of real springs and solid rectangular planes: this is not the case since the springs are massless and dimensionless, and images can occupy the same region as other images and pass unhindered through each other.

The number of different possible equilibrium positions depends on the configuration of the network and the initial positions of the images. To ensure that the simulation does not stop at a local minimum, constraints need to be introduced into the model. These are discussed in the following sections.

3.2 The Hinge Constraint

Based on the relationship between the sensor plane and the image position, a constraint can be used to impose this relative position on a pair of overlapping images. The constraint designed is a *non-rigid hinge*, defined, for every pair of overlapping images A and B, by two points, P_{H1A} and P_{H2A} , in image A which are attracted to the corresponding two points, P_{H1B} and P_{H2B} ,

in image B, as shown in Fig. 4. The attraction is a force whose magnitude depends on the distance between corresponding points. In Fig. 4(a), the two images are correctly aligned and the hinge points coincide ($\mathbf{P}_{H1A} \equiv \mathbf{P}_{H1B}$ and $\mathbf{P}_{H2A} \equiv \mathbf{P}_{H2B}$), whilst Fig. 4(b) shows the images separated, and forces \mathbf{F}_1 and \mathbf{F}_2 pull the images to the correct positions.

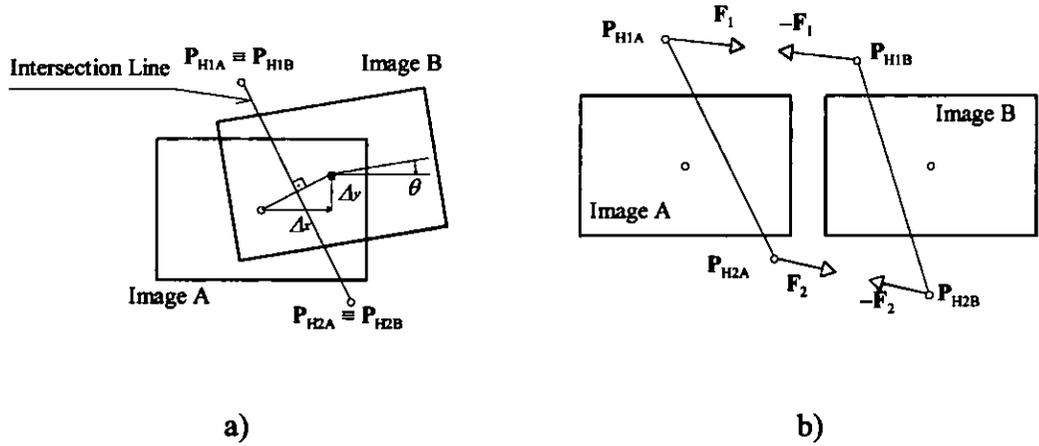


Figure 4 Non-rigid hinge model

Newton's third law states *action and reaction are equal in magnitude and opposite in direction*. Hence, if \mathbf{F}_1 is exerted on \mathbf{P}_{H1A} , its reaction force $-\mathbf{F}_1$ will be exerted on \mathbf{P}_{H1B} . This can be implemented with forces proportional to the distance between hinge points:

$$\begin{aligned} \mathbf{F}_1 &= k_H (\mathbf{P}'_{H1B} - \mathbf{P}'_{H1A}) \\ \mathbf{F}_2 &= k_H (\mathbf{P}'_{H2B} - \mathbf{P}'_{H2A}) \end{aligned} \quad (2)$$

For any pair i of neighbour images, the measure of the error between their current position in space and their actual registration is given by

$$E_i = \frac{|\mathbf{P}'_{H1A_i} - \mathbf{P}'_{H1B_i}| + |\mathbf{P}'_{H2A_i} - \mathbf{P}'_{H2B_i}|}{2} \quad (3)$$

which is the sum of the Euclidean distances between corresponding hinge points, where P'_{H1A_i} , P'_{H1B_i} , P'_{H2A_i} and P'_{H2B_i} are the hinge points expressed in world coordinates. This error is an indication of the image quality at the seam.

The maximum error, which is the error for the worst case of image pairs, is given by

$$E_{\max} = \max \{E_1, E_2, \dots, E_M\} \quad (4)$$

where M is the number of neighbour pairs of images, and the average error is given by

$$E_{\text{aver}} = \frac{\sum_{i=1}^M E_i}{M} \quad (5)$$

The simulation is run until the kinetic energy is totally dissipated. Some potential energy will remain in the system unless all the alignments were error-free. The quality of the resulting distribution of the images can be assessed by looking at the maximum error and the average error.

3.3 The Spherical Constraint

The *spherical constraint* is designed to induce the spherical distribution of images in the system without having to provide information regarding the radius of the sphere. The spherical constraint effectively makes the images face the same point, which, for convenience, is chosen to be the centre of mass of the system (CM_{SYS}). The constraint is implemented with two forces of the same magnitude, F_{CM} , and opposite direction (see Fig. 5) which rotate the

image towards the centre of mass of the system. The forces act on points P_{CM1} and P_{CM2} , which are situated at a fixed distance in front and behind the centre of the image respectively.

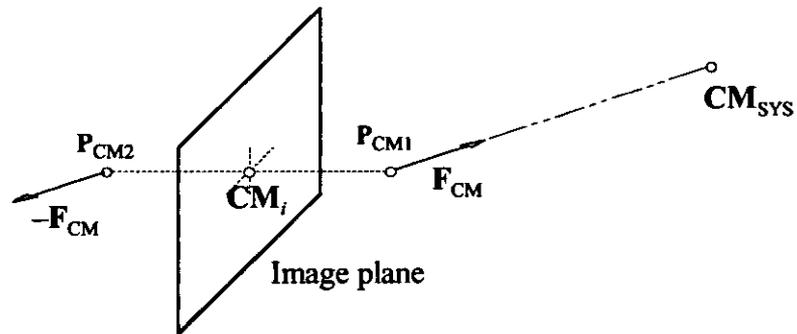


Figure 5 Spherical constraint

The magnitude of the force F_{CM} is proportional to the distance between the centre of the image and CM_{SYS} . The constant of proportionality is k_{CM} .

$$F_{CM} = k_{CM} (CM_{SYS} - CM_i) \quad (6)$$

4 EXPERIMENTAL RESULTS

4.1 Image Acquisition

The images in this experiment were obtained using a still digital camera connected to a computer. The scene is a laboratory in the University of Central Lancashire. The path followed was carefully planned in advance according to the following requirements:

- To ensure that the images cover the entire omnidirectional view, i.e. without leaving gaps in the mosaic.

- To provide sufficient overlapping between adjacent sections of the path for accurate registration.
- To avoid redundant images, i.e. too many images from the same regions of the scene.
- To ensure that sufficient texture is present in every pair of neighbour images, several small yellow notes were laid on the carpet to provide a reference for the registration, and a sign reading “HELLO” was stuck to the ceiling.
- To avoid objects too close to the camera that may appear defocused.

The trajectory of the camera is represented in Fig. 6. Images of the floor directly under the camera could not be taken.

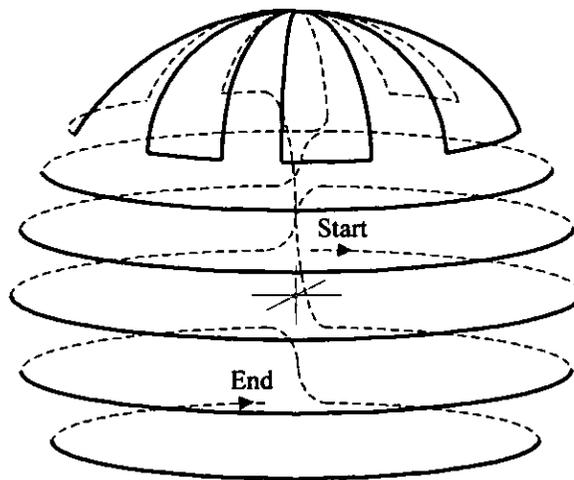


Figure 6 Trajectory of the camera

4.2 Simulation of the Physical Model

The list of neighbour images is initialised with successive image pairs. 24 extra neighbour pairs were added to the list in order to link both the bands and the images at the “HELLO” sign on the ceiling.

The hinge constraint alone does not encourage a spherical distribution of the images. As an example with synthetic data (of a perfect single band panoramic loop), Fig. 7 shows one of the many possible distributions satisfying the hinge constraint. The image planes are colour coded.

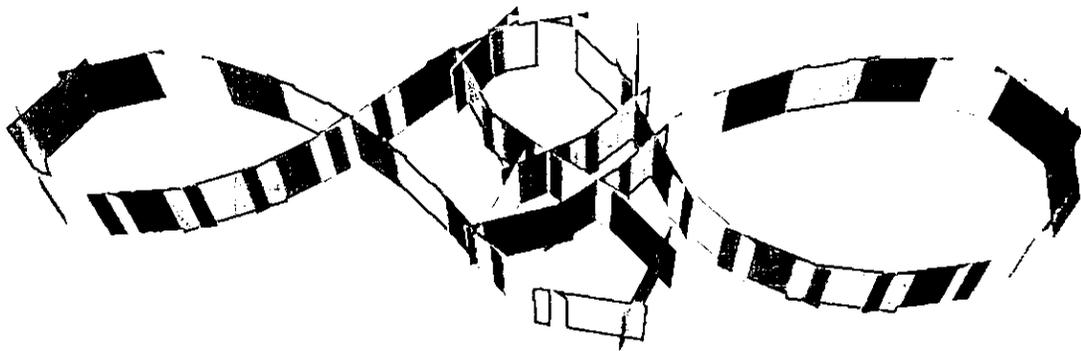


Figure 7 One of the possible distributions when only the hinge constraint is present

If the spherical constraint is applied to every image in the example in Fig. 7, it was found that the new forces can drive the system into a cylindrical distribution, as shown in Fig. 8.

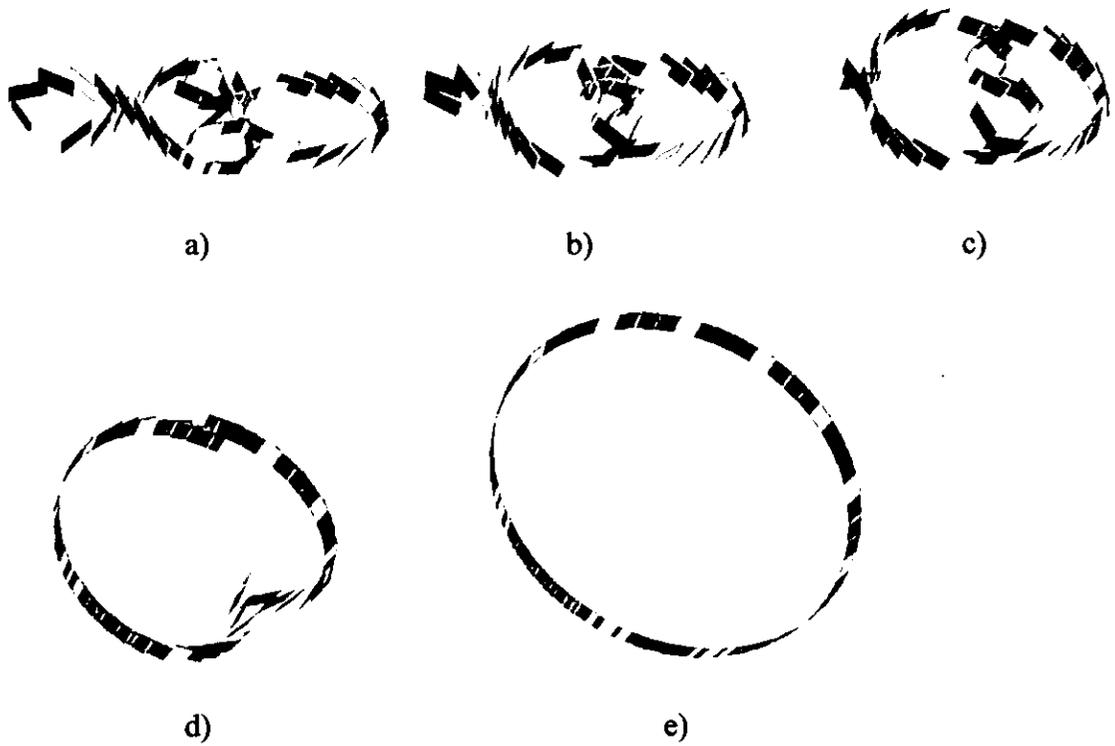


Figure 8 Sequence showing evolution of the image planes represented in Fig. 7 after introduction of the spherical constraint

The value of the constant k_{CM} must also be chosen carefully. It is not rare that a double loop or a figure of eight develops from the initial position in the plane (see Fig. 9). A high value of k_{CM} (with respect to the hinge constant k_H) would prevent the images from turning away from CM_{SYS} , consequently the loop cannot be 'untied', resulting in the stable distribution shown in Fig. 10. If, on the other hand, the value of k_{CM} is too low with respect to k_H , the progression to disentangle the loop becomes very slow and eventually halts completely.



Figure 9 A double loop in the image distribution (shown from two different points of view)



Figure 10 Typical final distribution of images for a high value of k_{CM}

4.3 Full Spherical Mosaic

An experiment was conducted to demonstrate the proposed mosaic construction method. The images are distributed approximately on the surface of a sphere. The sphere would be complete except for the missing images under the camera. Figures 11 to 13 show the evolution towards equilibrium of the system of images initially placed sequentially on a plane. The path followed by the camera over the scene is discernible in Fig. 11. The small white sphere in the centre of the right image in Fig. 13 represents the position of CM_{SYS} .

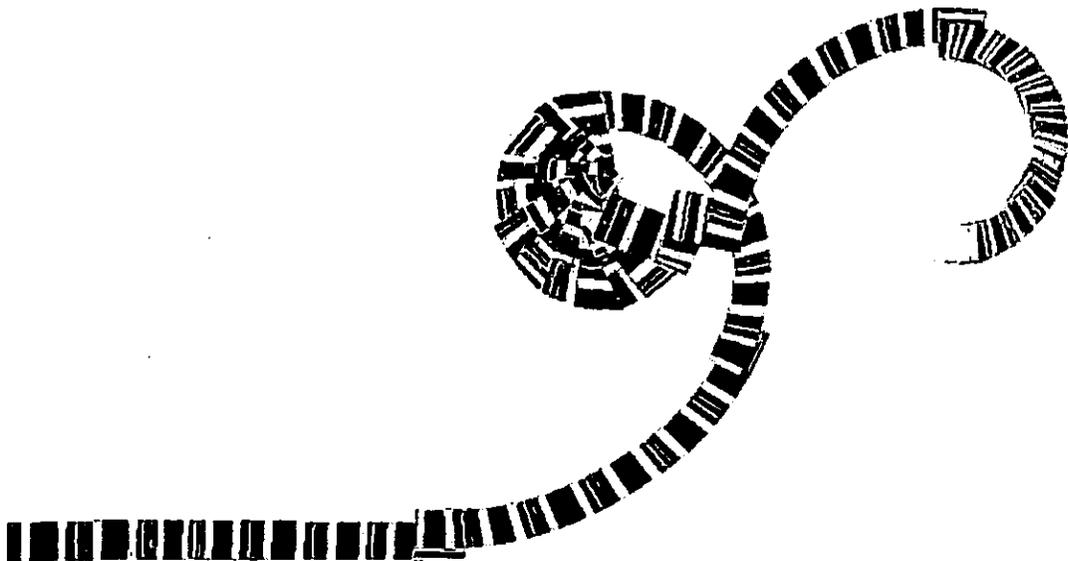


Figure 11 Initial position with images placed sequentially on a plane

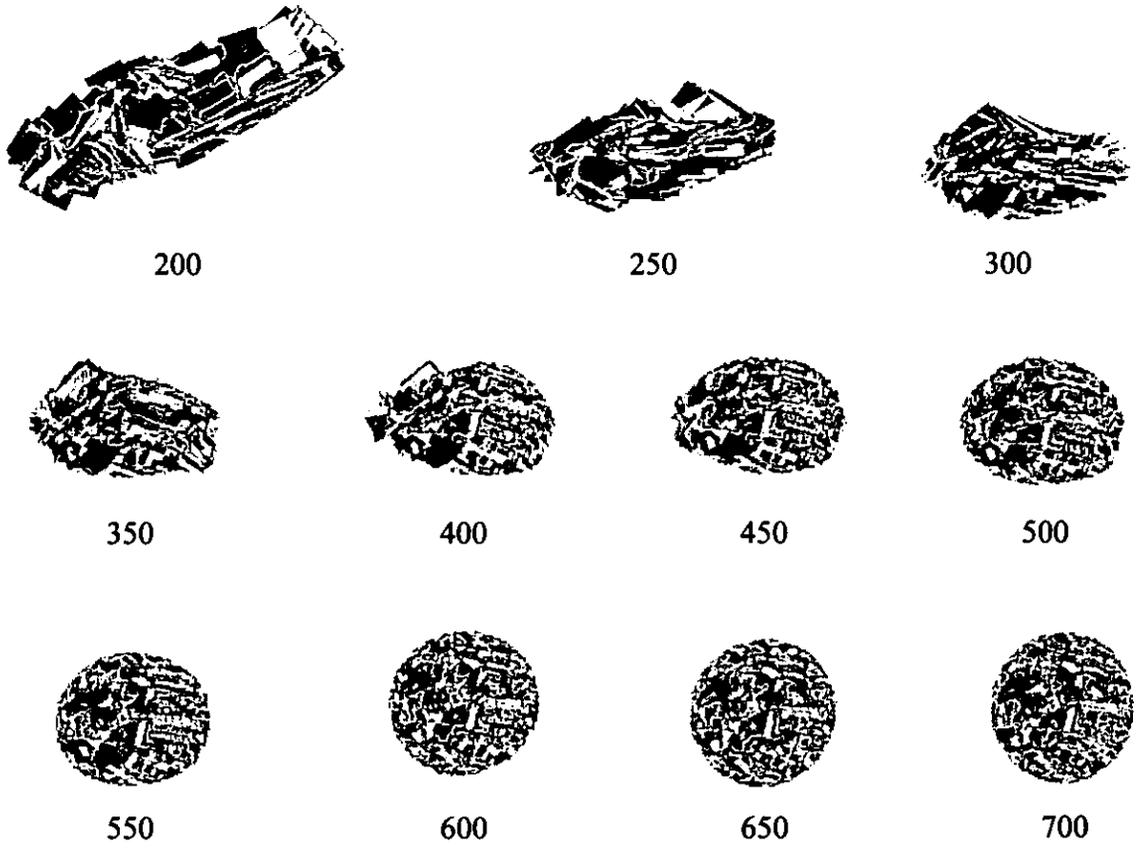


Figure 12 Evolution of the images towards a spherical shape (iterations 200 to 700)

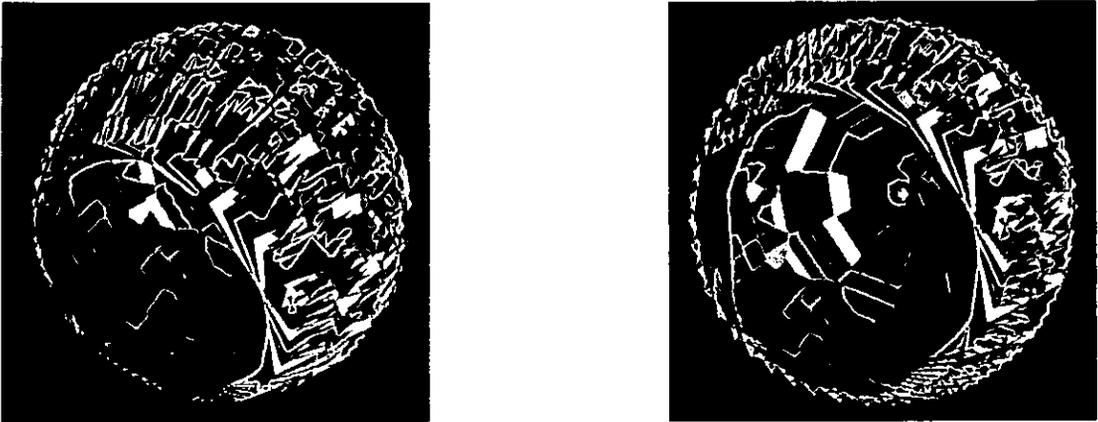


Figure 13 Two views of the final shape at iteration 900

4.4 Image Integration and Display

Once the positions of the images are known, it is necessary to find a practical means of representing the mosaic which can be quickly rendered or viewed. One approach [11] is to map the mosaic onto an arbitrary texture-mapped polyhedron surrounding the origin, this way the virtual environment can be explored using standard 3D graphic viewers and hardware without requiring special-purpose players. A cube with six separate texture maps was chosen for simplicity and for efficiency when using standard 3D graphic rendering engines. For each side of the cube, for each pixel in the texture image, a ray is traced from CM_{SYS} through that pixel. From the intersected image nearest to CM_{SYS} , the pixel value at the intersection of the ray with this image will be copied to the pixel in the texture image. This scheme corresponds to the *Voronoi tessellation*, which, given the position of the centres of the images, defines a polygonal area for each image to be pasted on the mosaic. The seams between adjacent images were blended using a weighted average of the pixels near to the seam [4]. Although simple, this considerably improves the perceived quality of the mosaic.

4.5 Displaying Results

A VRML (Virtual Reality Modelling Language) model was written containing the geometry of the cube and the references to the texture image files. An internet browser with a VRML plug-in can display the resulting mosaic. The user is virtually placed in the centre of the cube, and can look in any direction. VRML allows linking of selected areas on the mosaic to other web pages or other spherical mosaics, offering the possibility to create virtual scenarios with linked omnidirectional views. Fig. 14 shows the spherical mosaic as viewed by an internet browser with a VRML plug-in, and the six sides of the cube are shown in Fig. 15.

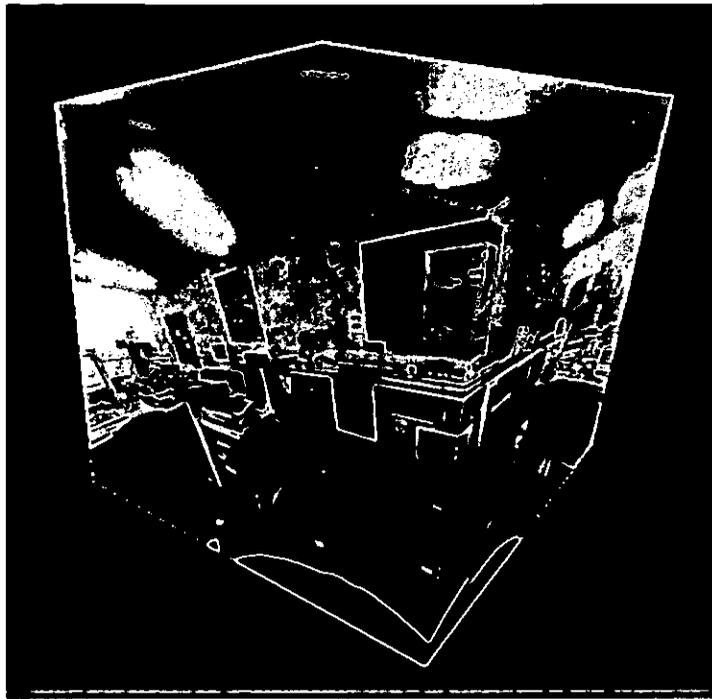


Figure 14 Spherical mosaic viewed using internet browser (Netscape) with VRML plug-in
(Cosmo Player)

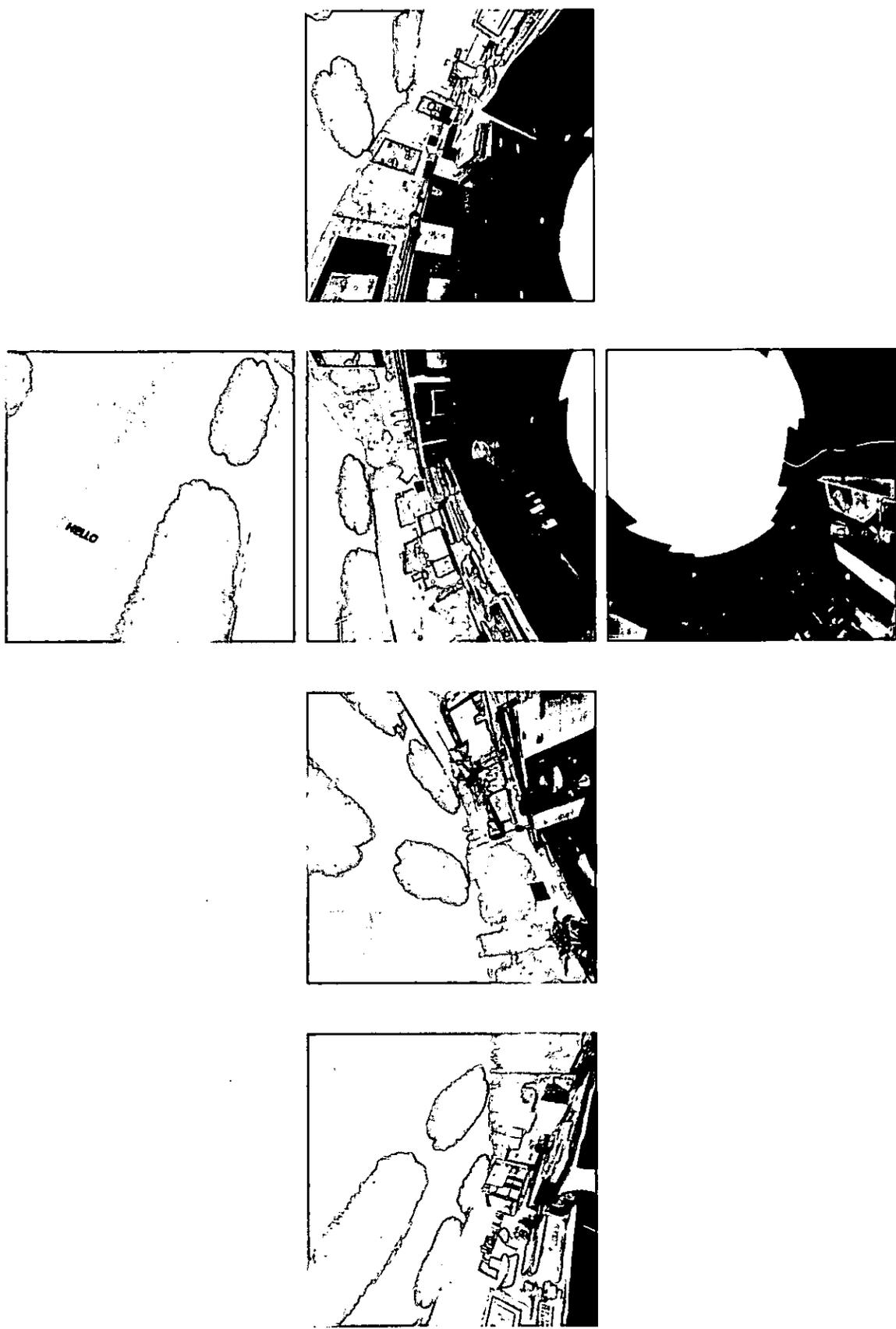


Figure 15 The six sides of the cube ready to be rendered

5 CONCLUSIONS

This paper describes the construction and display of a spherical mosaic from a sequence of overlapping digital images. A solution to the looping path problem has been found, and the hinge method proposed as a means of minimising alignment errors between neighbour images.

This paper introduces the use of an analogy between the images and a physical model. The images correspond to rigid objects, which are linked by forces which pull them towards the correct positions with respect to their neighbours. Constraints may be introduced in order to achieve a cylindrical or spherical distribution of the images. This network of self-organising images is implemented in a computer simulation.

As a direct result of the new concepts developed in this research work, a spherical mosaic was constructed which can be viewed from a web browser with VRML capabilities.

REFERENCES

- 1 Wolf, P.R., "Elements of photogrammetry (with air photo interpretation and remote sensing)", McGraw-Hill, New York; London, 1974
- 2 Horii, S., Oshima, Y. and Hirao, K., "Digital Image Processing", *Eighteenth International Symposium on Remote Sensing of Environment*, pp1785-1794, 1995

- 3 Milgram, D.L., "Adaptive Techniques for Photo Mosaicing", *IEEE Transactions on Computers*, vol. C-26, pp1175-1180, 1977
- 4 Peleg, S., "Elimination of seams in photomosaics", *Computer Graphics and Image Processing*, 16:90-94, May 1981
- 5 Burt, P. and Adelson, E.H., "A Multiresolution Spline with Application to Image Mosaics", *ACM Transactions on Graphics*, vol. 2, pp217-236, 1983
- 6 Szeliski, R., "Image Mosaicing for Tele-Reality Applications", *Cambridge Research Lab Technical Report*, CRL 94/2, May 1994
- 7 Guillén, M., Holifield, P. and Varley, M., "Improved Mosaic Construction by Accumulated Alignment Error Distribution", *Proceedings of the British Machine Vision Conference*, vol. 2, pp377-387, 1998
- 8 Delaunay, B., "Sur la Sphere Vide", *Bulletin of Academy of Sciences of the USSR*, pp793-800, 1934
- 9 Gümüstekin, S. and Hall, R.W., "Mosaic Image Generation on a Flattened Gaussian Sphere", *Third IEE Workshop on Applications of Computer Vision*, Florida, pp50-55, December 1996
- 10 Mann, S. and Picard, R.W., "Virtual Bellows: constructing high quality stills from video", *IEEE International Conference on Image Processing*, pp363-367, 1994
- 11 Szeliski, R. and Heung-Yeung, S., "Creating Full View Panoramic Mosaics and Environment Maps", SIGGRAPH, 1997