# Image Registration and its Application to Computer Vision: Mosaicing and Independent Motion detection

Ntana Nkanza

Submitted to the Department of Electrical Engineering,
University of Cape Town, in fulfillment of the requirements
for the degree of Master of Science in Engineering.

Cape Town, June 2005

# Declaration

I declare that this dissertation is my own work. It is being submitted for the degree of Master of Science in Engineering at the University of Cape Town. It has not been submitted before for any degree or examination at this or any other institution.

. . . . . .

Ntana Nkanza

(Candidate's Signature)

# Abstract

Image registration enables the geometric alignment of two images and is widely used in various applications in the fields of remote sensing, medical imaging and computer vision. This thesis explores each of the stages, and looks at two applications of image registration. The applications investigated are mosaicing and independent motion detection.

Mosaicing is the aligning of several images into a single composition that represents part of a 3D scene. This is useful for many different applications, including virtual reality environments and movie special effects.

Motion detection often assumes a static background onto which moving objects provide a dynamic foreground. A challenging problem is presented when the camera is also moving. A differentiation needs to be made between the apparent movement of the background (caused by the motion of the camera) and independently moving objects in the scene. We compensate for the apparent movement of the background by registering or aligning the images. Following this, we use frame-differencing, thresholding, and morphological operations to segment independently moving objects in the scene. We do not set out to achieve real-time motion detection, but rather present a means for the detection of independent motion.

The images we consider in this thesis are views of a scene taken by a rotating camera and are registered by means of a planar homography. The advantage of our approach to the two applications is that it is fully automated, robust and uses only information provided within the images. Mosaicing results show that provided there is sufficient overlap between images, mosaicing can be achieved with no user intervention. Lastly, it is shown that despite the apparent movement of the background it is indeed possible to detect independently moving objects using our approach.

# Acknowledgements

First and foremost I would like to give honor to God. “All things are possible to him who believes” (Mark 9:12), “I can do all things through Him who strengthens me” (Philippians 4:13). Through you everything is indeed possible and you have brought me so far.

To my two supervisors, Dr Fred Nicolls and Prof. Gerhard De Jager, I am thankful to you for affording me the opportunity to pursue such a stimulating and thought provoking research topic. It was a pleasure working with and learning from you. Without your guidance finishing this thesis would have been impossible. Also, I would like to thank DebTech, a division of De Beers, and the National Research Foundation for their financial support. Collin Andrews, your support through out this journey will not be forgotten.

To all the Digital Image Processing students, particularly Simphiwe, Keith, Phillip and Bruno, you created such a pleasant work environment and the advice you gave me was invaluable. Nicholas, Sifiso, Sonny, Valerian and Samson (being ahead doesn’t mean you finish first!), I appreciate you all being there when times got rough. To Aisha and Evonne (Evonne and Aisha?) you guys have played a huge role in my life and I will never forget that. You are the reason my years here have been special, I’ll miss all the laughs we had, sorry for all the tears.

Finally to my parents and my Uncle Neil, you have been amazing. Thank you so much for getting me here. Nobody has been there for me like you have and I dedicate this thesis to you.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

"If we knew what it was we were doing, it would not be called research, would it?"

- Albert Einstein (1879 - 1955)

*This thesis looks at the stages of image registration. It also considers the problems of creating image mosaics and the detection of independent motion as viewed by a non-stationary camera. The chapter starts with an overview of the research and application areas that provide inspiration to this work. Some selected work is briefly described, as relevant related work will be referred to within the individual chapters of this thesis.*

## 1.1 Motivation and scope

Vision allows us humans to observe and be aware of our surrounding world. Computer vision, as the name suggests, aims to duplicate human vision by electronically manipulating and interpreting images. Whilst images do provide us with lots of useful information, extracting this information is not a single problem but rather a vast number of them, each individually complicated. As such, computer vision has been evolving as a multi-disciplinary subject over the years focusing on the extraction, representation and use of visual information in artificial intelligence, robotics, medical image analysis, surveillance systems, and other applications.

Further, images not only contain scene shape, structure and colour information, but also the possible motion of the camera and objects in the scene. It is precisely this information that we manipulate in this thesis. Active vision allows for camera motion and thus improves the retrieval of information relevant to a particular task. For instance, in surveillance systems, active tracking can be used to keep an object of interest in sight, or better yet, keep it centred in a video sequence.

When multiple images are captured from different viewpoints (or at different times) the images become distorted with respect to each other. Image registration or *alignment*[1] is the process of determining the optimal transformation matrix that results in the images being in spatial alignment [16]. The registration of images is used in a variety of applications and it is no surprise that research in the area of image registration has followed many avenues towards determining this optimal transformation matrix. Excellent reviews of image registration are given in [4] and [58]. The reader is encouraged to look at these for an in-depth look at the different methods used in the registration of images.

This thesis considers each stage of image registration in depth and then looks at two application areas, namely the creation of mosaics and the detection of independent motion. The objectives of our research can be summarised as to:

- Review existing literature on image registration and tackle each of the stages involved in this process.
- Use image registration to create mosaics of scenes containing moving objects.
- Combine image registration and temporal differencing to achieve independent motion detection.
- Test the selected method of mosaicing and motion detection on images of real scenes.
- Finally, draw conclusions and make recommendations for future research directions.

## 1.2 Related work

Algorithms that allow images to be aligned and seamlessly stitched together are among the oldest and most widely used in computer vision [48]. One of the applications of image registration is in the medical field: [28] offers a comprehensive survey. In medical image analysis, image registration is used for applications ranging from tumour detection to those dealing with the integration of structural information from computed tomography (CT) or magnetic resonance (MR), with functional information from scanners such as Position Emission Tomography (PET) [4]. In the

[1] The terms registration and alignment will be used synonymously throughout this thesis.

field of remote sensing, image registration can be used for interpreting changes in scenes captured at different times, for instance in urban growth monitoring [9] or surveillance of nuclear plants. Other applications in the field of remote sensing include the location of positions and orientations of well known features such as parking lots, airport runways, etc. More information on the applications of image registration in the field remote sensing can be found in [4]. One last useful application is that of the creation of panoramic mosaics. Several approaches are presented in literature to construct full view panoramas by taking several video images in order to cover the whole viewing space and then stitch the images together. The reader is encouraged review work done by Szeliski (see credits). Other work related to this thesis includes [1, 7, 8, 11, 14, 17, 23 and 32].

## 1.3 Thesis structure

The rest of this thesis is organised as follows: firstly an introduction to projective geometry is given (Chapter 2) after which the feature detection and feature matching stages of the image registration are then explained. Here, a general overview of the stages and related literature is given before the exact methods that were used are described (Chapter 3). A background description of the estimation of the transform used in this research is given and the robust estimator that is used to estimate it described. The method used to align pairs of images is also presented (Chapter 4). Two applications of image registration are investigated; image mosaicing (Chapter 5) and motion detection (Chapter 6). Conclusions based on the research are drawn and directions for future research given at the end of the thesis (Chapter 7).

# Chapter 2

# Projective Geometry

"Copy from one, it's plagiarism; copy from two, it's research."

- Wilson Mizner (1876 - 1933)

*This chapter reviews some of the basic notations and properties of projective geometry. Projective geometry is the natural mathematical framework used to describe the projection of a scene onto an image. The background presented here is therefore useful for understanding subsequent chapters.*

## 2.1 Perspective camera model

The most commonly used geometric model of a camera in computer vision is the pin-hole camera. This model consists of a plane $R$ , called the *retinal* or *image* plane, and a 3D point $O$ called the *center of projection* or the *optical center*. The straight line through the optical center and perpendicular to the image plane is called the *optical axis* and the distance between the plane R and the optical centre, the *focal length*. The camera reference frame has its origin at the optical centre and the optical axis as its z-axis. This discussion is illustrated in figure 2.1.

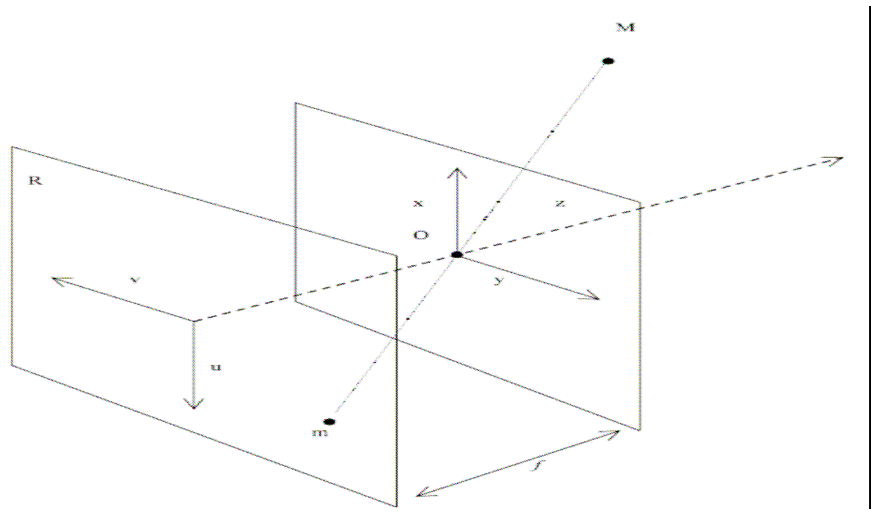



**Figure 2.1: The perspective Camera Model**

Let $(u, v)$ be the 2D coordinates of the point **m** and $(x, y, z)$ the 3D coordinates of **M**. The fundamental equations of the perspective projection of **M** in the point **m** of the image plane are:

$$u = f\frac{x}{z} \qquad v = f\frac{y}{z} \tag{2.1}$$

Equation 2.1 is a non-linear relation but can be expressed in homogenous coordinates and expressed linearly as:

$$\lambda\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \Leftrightarrow \quad \lambda\tilde{m} = P\tilde{M} \tag{2.2}$$

where P is the perspective projection matrix.

## 2.2 Camera parameters

Often, the camera reference frame is unknown and the location and orientation of the camera frame with respect to a known reference frame needs to be determined using image information only. In practice the image coordinate system is represented by a pixel grid (not mm) and the origin is not the principal point but one of the image corners. Also, the horizontal and vertical distance between pixels is not necessarily the same. To deal with these issues, an operation known as camera calibration is performed. This operation involves estimating two sets of camera parameters known as *extrinsic* and *intrinsic* parameters. Extrinsic parameters define the location and orientation of the camera reference frame with respect to a known reference frame. A typical choice for describing the transformation between the camera and world frame is to use a 3D translational vector **T** describing the relative positions of the two frames, and a rotational matrix R that brings the corresponding axes of the two frames onto each other. The relation between the coordinates of the point **M** in world and camera frame, $\mathbf{M}_w$ and $\mathbf{M}_c$ respectively, is:

$$\mathbf{M}_c = \mathrm{R}(\mathbf{M}_w - \mathbf{T}) \tag{2.3}$$

with

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$$

Intrinsic parameters are the parameters necessary to link the pixel coordinates of an image point with the corresponding coordinates in the camera frame and characterize the optical, geometric and digital characteristics of the viewing camera. They specify, respectively; the perspective projection (i.e. the focal length), the transformation between the camera frame coordinates (mm) and pixel coordinates, the geometric distortion introduced by optics. Neglecting geometric distortions, the transformation between camera and image frame coordinates is given by:

$$x = -(x_{im} - o_x)\, s_x \qquad y = -(y_{im} - o_y)\, s_y \tag{2.4}$$

where $(o_x, o_y)$ are the coordinates of the pixel of the image center $O$ (the centre of projection), and $(s_x, s_y)$ is the effective size of the pixel (in mm) in the horizontal and vertical direction respectively. Putting equations 2.3 and 2.4 into equation 2.1, and neglecting radial distortion, two matrices for the cameras intrinsic and extrinsic parameters may be defined as:

$$M_{\text{int}} = \begin{bmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$M_{ext} = [\mathrm{R};\mathbf{T}]$$

where [R;**T**] represents a $3 \times 4$ matrix with the first three columns occupied by R and the fourth by the vector **T**.

# 2.3 Planar transformations

A linear transformation of a projective space is defined by a non-singular $(n+1)\times(n+1)$ matrix A. This transformation is known as a collineation or a projective transformation. The matrix A performs an invertible mapping of onto itself and is defined up to a non-zero scale factor. For 2D projective transformations we have what is known as the projective plane and two different views of the same planar scene in 3D space are related by a collineation which is also known as *homography*, the term we adopt in the rest of this thesis.

A 2D projective transformation is a linear transformation on homogeneous 3-vectors represented by a non-singular $3\times 3$ homography matrix H:

$$\begin{bmatrix} x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{2.5}$$

In equation 2.5, $(x_1, x_2, x_3)$ and $(x_1', x_2', x_3')$ are the homogeneous vector representations of two points, and H is the matrix defining the linear mapping of homogeneous coordinates. The matrix H may also be multiplied by any arbitrary non-zero scaling factor without altering the projective transformation. The matrix H has eight degrees of freedom, being defined up to a scale factor. Two images taking by a moving camera are related by a homography if the scene is planar or if the point of view of the camera does not change (the camera is rotating around its optical axis). Table 2.1 shows the hierarchy of homographies.

**Table 2.1: Models used to describe a planar transformation.**

| Image model and degrees of freedom | Homography Matrix form | Distortion and invariants |
|---|---|---|
| Pure translation<br>2 DOF | $\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$ | Image is parallel to the planar scene. No rotation. |
| Euclidean<br>3 DOF | $\begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$ | Translation and rotation are distorted. Length is invariant |
| Similarity<br>4 DOF | $\begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$ | Translation, rotation and scale are distorted. Angle is invariant. |
| Affine<br>6 DOF | $\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix}$ | Translation, rotation, scale and shear are distorted. Parallelism is invariant |
| Projective<br>8 DOF | $\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$ | Most general planar transform, collinearity and cross-ratio are invariant |

Notably, a general projective transformation takes into account translation, rotation, scaling, shear, and perspective deformation. When dealing with a 3D scene and arbitrary camera motion, the relationship between two views can be defined in terms of a homography plus a parallax term depending on the scene structure and camera translation. However if the depth range of the scene is small compared to the distance from the camera, or the translation is small, then the parallax term can be neglected [12].

## 2.4 Conclusions

This chapter was devoted to a summary of the mathematical framework describing the projection of a scene onto an image. The perspective camera model, which is widely used in computer vision applications, was described. Following this, the cameras external and internal parameters were discussed. The process of finding the camera parameters, camera calibration, was not described as it fell beyond the scope of our research. However, for applications such as 3D reconstruction and motion tracking, camera calibration is essential. Finally, the basic properties of 2D projective transformations were explained. Further details on the computation of the homography will be discussed in subsequent chapters.

# Chapter 3

# Feature Detection and Matching

"It is impossible to begin to learn that which one thinks one already knows."

- Epictetus (c.55 - c.135)

*This chapter is devoted to the first two stages of image registration, namely feature detection and matching. The material presented here is not new and a plethora of literature on the subject exists. For more details on these topics the reader is referred to [4]. The chapter is organized as follows: Section 3.1 gives an introduction and describes the various types of features that can be used for our application. The features that are used in this thesis are corners. A review of corner detection and a description of the method we use are presented in section 3.2. The matching procedure is described in section 3.3 and the results of both the detection and matching shown in section 3.4. The chapter concludes with a discussion in section 3.5.*

## 3.1 Introduction

The feature detection and feature matching stages of image registration can be divided into two approaches, namely area-based and feature-based. Area-based methods of feature detection put emphasis on the matching of the features themselves rather than on their detection. These methods are not considered here; instead we focus on feature-based methods.

Feature-based methods do not work directly with image intensity. Therefore when illumination changes are expected, as is the case with the real scenes being dealt with in this research, the fact that features represent information on a higher level motivates their use. The use of feature-based methods is also recommended if the images contain sufficiently many detectable and distinct objects, as is the case with most computer vision applications. They are several features that may be used for

detection and matching, and certain criteria are used to justify the type of feature chosen. These criteria are that the features should be unique, able to be detected without difficulty, and have a good spatial distribution over the images. Some of the features that may be used in the registration of images, taken from [58], follow.

One type of feature that may be used is a closed-boundary region. Examples include buildings, forests, and fields or lakes due to their significant size in remotely sensed images. Regions are also used because their centres of gravity are invariant with respect to rotation, scaling and skewing, and are stable under conditions such as noise and grey level variations. However, region features are detected by means of segmentation methods, the accuracy of which significantly influences the resulting registration [34]. Line features such as representations of general line segments or object contours may also be used. Often the line correspondences are expressed by pairs of line ends or middle points [58]. Standard edge detection methods are used for line feature detection, and these include the Canny edge detector and the Laplacian of Gaussian (LoG) based detector. The reader is pointed to [57] where an extensive overview of edge detection techniques is given.

The last group of feature we consider are point features. This group of features includes methods using line intersections, high variance points, maximally distinct points with respect to a specified measure of similarity, and corners. With regard to feature detection, in most instances the core algorithms follow the definition of a point as a line intersection or as the centroid of a closed-boundary region. It has been found that corners form their own class of feature as the property of being a corner is hard to define mathematically.

## 3.2 Corner detection

As mentioned in the previous section, there are several types of features available for detection in images. Primarily, the choice of which feature is to be used is dependent on the application being undertaken. Corners, with their two-dimensional structure providing information about image motion, are well suited to the moving scenes being considered in this research. The choice of corners for feature detection is further motivated by the fact that they are stable, not only to small changes in viewing

directions but also to illumination changes. Also, corners are not affected by the aperture problem inherent when computing optical flow.

A review of some of the different methods of corner detection is now given, after which the corner detector used in this research is described and reasons for its choice given.

### 3.2.1 Review of corner detectors

Early approaches to corner detection involve segmenting images into regions, extracting boundaries as chain codes, and then identifying corners as points where directions change rapidly [38]. Such approaches have largely been done away with due to their dependence on the initial segmentation step. More recent corner detectors can be categorized in two groups that differ in the way "cornerness" is defined. We have curvature-based detectors, as well as feature-based detectors.

The curvature-based corner detectors exploit the definition of a corner as a point where the edge contour curvature is high. Kitchen and Rosenfeld [25] employ a local quadratic surface to find the magnitude of the gradient and the rate of change of the gradient direction. The resulting product of these two quantities is determined and the point of local maximum locates the corners. Baudet [3] improves high-curvature edges by looking for saddle points in the image brightness surface, and then calculates the image Gaussian curvature based on the product of the two principle curvatures. The Baudet operator, known as the DET, is derived from the second-order Taylor expansion of the intensity function $I(x, y)$:

$$DET = I_{xx} I_{yy} - I_{xy}^2 , \tag{3.1}$$

where $I_{xx}$, $I_{yy}$ and $I_{xy}$ are the second order partial derivatives of $I(x, y)$. The corner detection is based on the thresholding of the maximum of this cornerness measure.

At the forefront of the curvature-based methods is the Wang and Brady corner detector [53]. Here the stability of the detected corners is improved by suppressing false corners that are wrongly reported on strong edges. The original image is convolved with a Gaussian filter ($\sigma$=0.5 pixels) to reduce the effect of noise and

quantization, before computing the total image surface area. It is shown that for points with a strong gradient, the total curvature κ can be approximated by

$$\kappa^2 = \left( \frac{\frac{\delta^2 F}{\delta t^2}}{|\nabla F|} \right)^2 > S\,, \tag{3.2}$$

which leads to

$$\left( \frac{\delta^2 F}{\delta t^2} \right)^2 - S|\nabla F|^2 > 0\,, \tag{3.3}$$

where F is the grey-level image after Gaussian convolution and $\delta^2F/\delta t^2$ is a directional derivative along the direction perpendicular to the image gradient **n.** The term $S|\nabla F|^2$ is the edge strength, which responds well at the edge pixels.

A modified corner detector is proposed in [53] which looks for where the curvature κ is high and where a local maximum is found in the inequality given in equation 3.3. The Wang and Brady detector is therefore defined as:

$$\Gamma = \left( \frac{\delta^2 F}{\delta t^2} \right)^2 - S|\nabla F|^2 = Maximum \tag{3.4}$$

$$\frac{\delta^2 F}{\delta n^2} = 0 \tag{3.5}$$

$$|\nabla F|^2 > T_1, \Gamma > T_2 \tag{3.6}$$

where S is a constant measure of the image surface curvature (varying with different Gaussian masks), F is the intensity image after Gaussian smoothing, and $T_1$ and $T_2$ are user-defined thresholds on edge and corner strengths. In cluttered environments, however, the false corner suppression is not sufficient to prevent false responses on strong diagonal edges. This group of detectors is sensitive to noise as the measure of "cornerness" relies on the second order derivatives.

Feature point based detectors use the intuitive definition of a corner as points that are well-distinguished from neighbouring points, or where the local autocorrelation of the image intensity is high. Paler et al. [35] show that, at a corner, the median of the local brightness values taken over a small mask is significantly different from the centre

value. A corner response is produced using the difference between the median and centre value of the mask. The approach is restricted to images where the edge widths and the contrast between the object and background can be estimated accurately.

Smith and Brady [45] introduce the SUSAN corner detector for low-level image processing, and use a principle similar to that of Paler et al. Their method considers an arbitrary pixel in an image and a corresponding circular pixel mask around it, the centre of which is called the nucleus. Provided that the image is not textured, there exists a compact region within the pixel mask whose pixels have similar intensities to the nucleus. The area is called the USAN (Univalue Segment Assimilating Nucleus), and by observing how the position of the centre of gravity of the USAN varies from the nucleus, a principle is derived to locate corners. Some of the representative shapes of the USAN are shown in Figure 3.1. No assumption is made about the form of the local image structure around any well-localised point, nor are points of interest sought. As such the SUSAN detector is fast and able to handle all types of junctions.

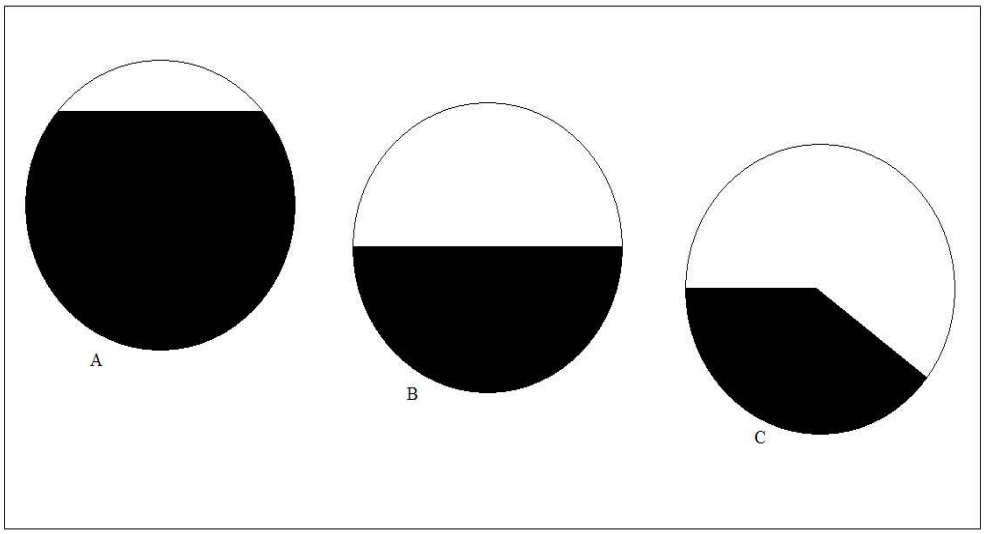



**Figure 3.1: Representative Shapes of USAN.** (A) the nucleus is within the USAN; (B) the nucleus is an edge point; (C) the nucleus is a corner point.

The Moravec corner detector [29] defines corners as points where there is a large intensity variation in every direction. The principle consists of computing an un-normalised local autocorrelation in four directions and taking the lowest results as the intermediate response. The final response is obtained after performing thresholding and local non-maximal suppression. As only four directions are used in finding the

local autocorrelation, the Moravec detector is sensitive to strong edges under certain directions. Harris and Stephens [18] use a similar technique to that of Moravec but estimate the local correlation measurements from first order derivatives. This is the detector used in this research and is the focus of the next section.

### 3.2.2 Corner detection using the Harris Corner Detector

To be useful for the later feature matching stage, a corner detector needs to satisfy the following criteria of robustness:

- **Consistency of detection -** The corner detector should detect even very subtle corners, while being insensitive to the variation of noise,
- **Localisation -** The corners should be detected as close as possible to their correct locations,
- **Stability -** The detected positions of corners should not move when multiple images of the same scene are acquired possibly from different viewpoints, and
- **Complexity -** For real-time tasks the corner detection needs to be fast, so a low algorithm complexity is required.

The Harris and Stephens corner detector is a widely used corner detector and is used in view of the above-mentioned criteria. Consider a local window in the image. Harris and Stephens determine the average variation in intensity that results from shifting the window by a small amount in different directions. Letting $I$ denote the image intensities and $W$ specify the current image window, the change $E$ produced by a shift $(x, y)$ is given by

$$E_{xy} = \sum_{u,v} W_{u,v} \left[ I(u+x, v+y) - I(u,v) \right]^2 . \tag{3.7}$$

An expansion about the shift origin is then performed as:

$$E_{xy} = \sum_{u,v} W_{u,v} \left[ xI_x + yI_y + O(x^2, y^2) \right]^2 \approx (x,y) M (x,y)^T , \tag{3.8}$$

where

$$M = \begin{pmatrix} I^2{}_x \otimes W & (I_x I_y) \otimes W \\ (I_x I_y) \otimes W & I^2{}_y \otimes W \end{pmatrix}. \tag{3.9}$$

$I_x$ and $I_y$ are the first order derivatives of the intensity function in the $x$ and $y$ directions respectively, $O(x^2, y^2)$ represents higher order properties, and $W$ is defined as the Gaussian function for a smooth circular window. Denote the

eigenvalues of the matrix M as α and β. Since the matrix M describes the shape of the local autocorrelation function at the origin of the shift, α and β are proportional to the principal curvatures and form a rotational invariant description of M. Therefore, when both α and β are larger than some threshold values, the shifts in any direction lead to a significant change in $E$ and a corner is flagged. The cornerness measure $C_S(M)$ is defined using the trace $Tr(M)$ and the determinant $Det(M)$ as:

$$C_S(M) = Det(M) - kTr^2(M)$$

where (3.10)

$$k = \frac{t}{(1+t)^2} \quad \text{and} \quad \frac{1}{t} < \frac{\alpha}{\beta} < t$$

The constant $k$ varies with different masks and different Gaussian convolution. As it is commonly used, the detection process used in this research may be summarised as consisting of the following stages:

- The calculation of the image gradients $I_x$ and $I_y$
- Convolution of the image gradients $I_x$ and $I_y$, and their product, $I_x I_y$ with a smooth circular Gaussian convolution mask
- Calculation of the corner responses from the smoothed gradients, and
- Thresholding the corner responses and applying a non-maximum suppression process to eliminate multiple candidates for a corner point

The results of using this corner detector are shown and discussed at the end of this chapter.

# 3.3 Corner matching

Once the two sets of corner features in the images have been detected, the aim is to match the corresponding features using spatial relations or various descriptions of the features.

## 3.3.1 Review of Feature-based Matching

Early work regarding feature based-matching was undertaken by Barnard and Thompson in [2]. In their approach, well-localised corners are found using the Moravec corner detector. An iterative relaxation of the matching surface is then used

to find an optimal set of matches. For each point in the first image a probability of a match is assigned to each point in the second, and relaxation is then applied in order to force the flow field to vary smoothly in the images. Shapiro et al [42] make use of a correlation method in their corner matching having found corners using the Wang and Brady detector. The corners are then matched using a correlation of small patches placed on the detected corners. Trajkovic and Hedley [52] use this same approach whilst working with the SUSAN corner detector. They use the standard cross-correlation coefficient for matching and a disparity constraint to reduce the number of mismatches.

When finding feature correspondences certain criteria need to be fulfilled by the feature description, namely:

- **Invariance -** the descriptions of corresponding features from both the images are required to be the same,
- **Uniqueness -** different features should have different descriptions, and
- **Stability -** the description of a feature, which is slightly deformed in an unknown manner, should be close to the description of the original feature.

An appropriate trade-off is often found between these conditions, and they are not all required to be satisfied. Features from the images with the most similar invariant description are paired as corresponding features. The selection of the types of the invariant description depends on the feature characteristics and also on the assumed deformation of the images. When searching for the best matching feature pairs in the space of the feature descriptors, the minimum distance rule with thresholding is normally applied.

### 3.3.2 Matching through correlation

In this research, the correlation and strength of match measure presented in the paper by Zhang et al. [56] are considered in the matching step of the image registration process. Consider an arbitrary corner $\mathbf{m}_1$ detected in the first of two images, with image coordinates given by vector $\mathbf{m}_1 = [u_i, v_i]^T$. The aim of corner matching is to find the corresponding feature $\mathbf{m}_2$ in the second image. As comparing $\mathbf{m}_1$ to all the corners in the second image is computationally expensive, the common approach is to use is a correlation-based technique.

Given a correlation window of size $(2n+1)\times(2m+1)$ centred at $\mathbf{m}_1$, a rectangular search area of size $(2d_u+1)\times(2d_v+1)$ is selected around this location in the second image. This search area has to be larger than the expected displacement of the feature between the two frames and a priori knowledge of the disparity between the matched points is needed. Searching for a match is reduced from the entire image to this area. Prior to the correlation, an image smoothed with an averaging filter of size $w\times w$ is subtracted from both images. This is done to compensate for any brightness differences in the images and allow a faster correlation calculation.

A correlation operation is then performed on a given window between point $\mathbf{m}_1$ and all the corners $\mathbf{m}_2$ lying within the rectangular search area in the second image. All the corners from the second image lying in this search area are considered candidates for the match and compared with $\mathbf{m}_1$. A score or a measure of similarity is computed between the small neighbourhood around the corner $\mathbf{m}_1$ and a correlation window in the neighbourhood of all the feature match candidates. This procedure is shown in Figure 3.2.

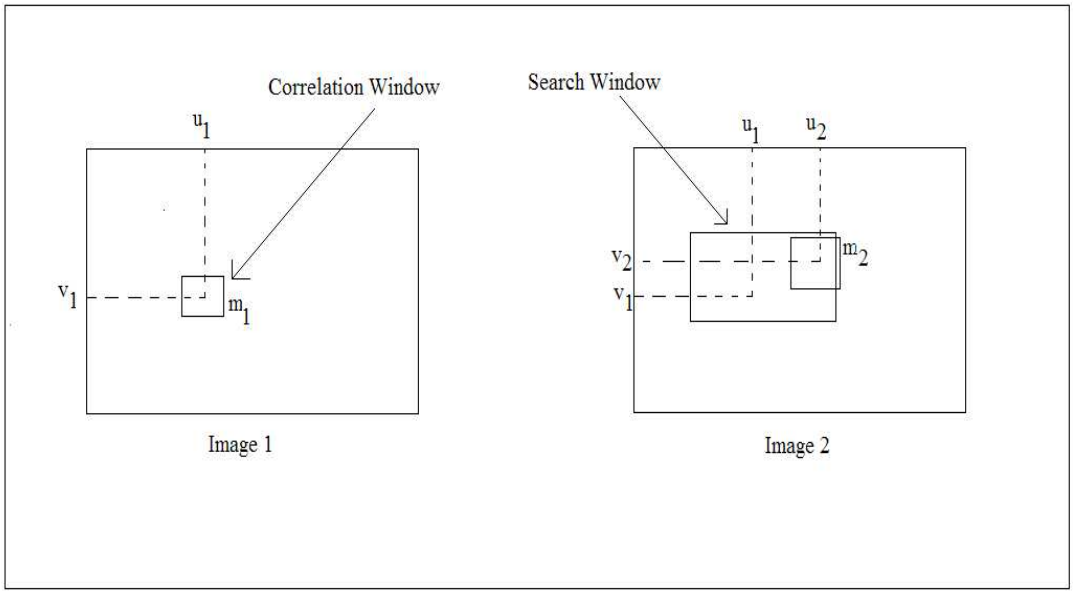



**Figure 3.2: Searching for a match by correlation.** Figure obtained from [56].

In our implementation, the *normalised cross correlation* (NCC) coefficient is the similarity measure and the correlation score is defined as:

$$\text{Score}(\mathbf{m}_1, \mathbf{m}_2) = \frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m}\left[I_1(u_1+i, v_1+j) - \overline{I_1(u_1, v_1)}\right] \times \left[I_2(u_2+i, v_2+j) - \overline{I_2(u_2, v_2)}\right]}{(2n+1)(2m+1)\sqrt{\sigma^2(I_1) \times \sigma^2(I_2)}}, \quad (3.11)$$

where

$$\overline{I_k(u,v)} = \frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m} I_k(u+i, v+j)}{(2n+1)(2m+1)}, \quad (3.12)$$

is the average at a point $(u,v)$ of $I_k$ $(k=1,2)$, and $\sigma(I_k)$ is the standard deviation of the image $I_k$ in the neighbourhood $(2n+1)\times(2m+1)$ of $(u,v)$, given by

$$\sigma(I_k) = \sqrt{\frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m} I_k^2(u,v)}{(2n+1)(2m+1)} - \overline{I_k(u,v)}}. \quad (3.13)$$

The score ranges from -1, for two correlation windows which are totally dissimilar, to 1, for two correlation windows that are identical (that is, for perfect correlation). In order to select the most consistent matches a constraint is placed on the correlation score. A match that has the highest correlation and scores above a predetermined threshold value is selected and forms a *candidate match*. Other measures of similarity that may be used in place of the NCC may be found in [46].

For each corner in the first image, we obtain a set of candidate matches from the second image and vice versa. Notably, it is possible to obtain no candidate matches for certain corners. A sufficiently large number of detected corners is needed to avoid this situation hampering the whole registration process. Further, the matching process is an ill-posed problem, and although the threshold on the similarity measure reduces the number of mismatches it is still impossible to eliminate the occurrence of mismatches.

Depending on image content and time considerations, different values may be used for the filter size, correlation window and other variables. In our implementation, as default values, w = 39 is used for the filter size, n = m = 7 is used for the correlation window size, and a threshold of 0.7 on the correlation score is chosen. For the search area, $d_u$ and $d_v$ are generously set to half of the image width and height, respectively.

This was done as we assumed the images overlapped by at least 50%. Also, whereas increasing the search area increases the probability of a bad match as there are more candidates, making the area too small is too restrictive

### 3.3.3 Support for a candidate match

The correlation method described in the previous section results in corners in the first image possibly having several candidate matches in the second image and vice versa. Matching ambiguities may be resolved in several ways. This section explains and defines the strength of the match as described in [56], and shows how putative matches can be obtained.

Consider a candidate match ($\mathbf{m}_{1i}$,$\mathbf{m}_{2j}$) where $\mathbf{m}_{1i}$ is a corner detected in the first image and $\mathbf{m}_{2j}$ is a corner detected in the second image. Let $\mathrm{N}(\mathbf{m}_{1i})$ and $\mathrm{N}(\mathbf{m}_{2j})$ be, respectively, the neighbours of $\mathbf{m}_{1i}$ and $\mathbf{m}_{2j}$ within a disc of radius R. If ($\mathbf{m}_{1i}$,$\mathbf{m}_{2j}$) is a good match, we expect to see many matches ($\mathbf{n}_{1k}$,$\mathbf{n}_{2l}$), where $\mathbf{n}_{1k} \in \mathrm{N}(\mathbf{m}_{1i})$ and $\mathbf{n}_{2l} \in \mathrm{N}(\mathbf{m}_{2j})$ such that the position of $\mathbf{n}_{1k}$ relative to $\mathbf{m}_{1i}$ is similar to that of $\mathbf{n}_{2l}$ relative to $\mathbf{m}_{2j}$. Alternatively, if ($\mathbf{m}_{1i}$,$\mathbf{m}_{2j}$) is a bad match, few or no matches are seen in their neighbourhood. The measure of support or strength for a match, $S_M$ is defined as:

$$S_M(\mathbf{m}_{1i},\mathbf{m}_{2j}) = c_{ij} \sum_{n_{1k} \in N(m_{1j})} \left[ \max_{n_{2l} \in N(m_{2j})} \frac{c_{kl}\,\delta(\mathrm{m}_{1i}, m_{2j}; n_{1k}, n_{2l})}{1 + dist(\mathrm{m}_{1i}, m_{2j}; n_{1k}, n_{2l})} \right], \qquad (3.14)$$

where $c_{ij}$ and $c_{kl}$ are the correlation scores of the candidate matches given in the previous section and $dist(\mathrm{m}_{1j}, m_{2j}; n_{1k}, n_{2l})$ is the mean distance of the two pairings, and

$$\delta(\mathrm{m}_{1i}, m_{2j}; n_{1k}, n_{2l}) = \begin{cases} e^{-r/\varepsilon_r} & \text{if } (n_{1k}, n_{2l}) \text{ is a candidate match and } r > \varepsilon_r \\ 0 & \text{otherwise,} \end{cases}$$

with the relative distance given by

$r = \dfrac{\left| d(m_{1i}, n_{1k}) - d(m_{2j}, n_{2l}) \right|}{dist(\mathrm{m}_{1i}, m_{2j}; n_{1k}, n_{2l})}$ and $\varepsilon_r$ a threshold on the relative distance difference.

Further details on this measure of support for a candidate match can be found in [56] and are omitted here.

With regard to Zhang's measure of support the following remarks are made in [21]:

- For the support of the match, only the candidate matches whose positions relative to the considered match are similar in the neighbourhoods are counted.
- The test of similarity in the relative positions is based on the relative distance, r. Whilst the similarity in relative positions is justified by the hypothesis of an affine transformation being able to approximate the difference between the small neighbourhoods of the proposed match, Zhang et al's criterion allows for a larger tolerance in distant differences for distant points.
- If a corner detected in the first image has several candidate matches in the second image, only the one, which has the smallest distance difference, is accounted for. This is achieved using the "max" operator.
- The contribution of each of the candidate matches is weighted by its distance to the match. A close candidate match thus gives more support to the match under consideration than a distant one.

One pitfall in this measure of support is that it is not symmetric. That is to say, it is possible that the strength of the match is not the same if the role of the images is reversed. This occurs when several corners in one image are candidate matches for the same corner in the other image as shown in Figure 3.3.

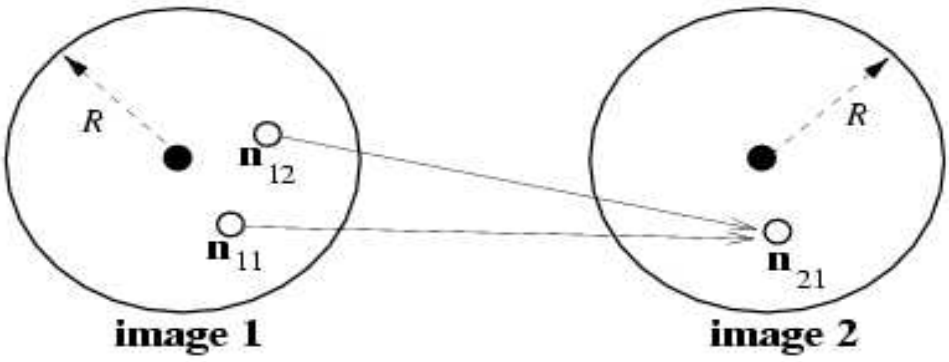



**Figure 3.3: Illustration of the non-symmetric problem of the measure of support.** $\mathbf{n}_{11}$ and $\mathbf{n}_{12}$ share the same corner $\mathbf{n}_{21}$ as a candidate match. Figure obtained from [56]

Before the summation is computed, if several corners score the maximal value with the same point, then only the corner which gives the largest value is counted. This is done to ensure that the same pairing is counted in the event of the role of the two images being reversed. In [56], a relaxation method is used to obtain putative

matches. This method consists of minimising an energy function that sums up the strengths of all the candidate matches. In our implementation, a similar approach is followed. A support of match matrix is formed whose rows represent the corners in the first image and columns represent the corners in the second image. The entries that are both highest in their respective columns and rows are chosen as matches.

Pilu [37] finds the putative matches via *Singular Value Decomposition (SVD)*. This approach is also followed in [21] where it is suggested that the relaxation method be replaced by, or combined with the SVD approach. However, the justification for this suggestion is that the SVD approach works well when only a few features are used. In our implementation however, due to the large number of features we use, the computational cost of using the SVD approach is high.

# 3.4 Experimental results

In this section, the results of the corner detection by the Harris detector and the matching by correlation are examined. The Harris detector and correlation matching are tested on the basis of:

- **Accuracy –** to test this criteria we use various types of images that would present the methods with a range of corner types, and
- **Stability –** by observing the matches obtained by the correlation-based matching. Better results can be obtained if a sub-pixel matching technique is used, but this comes at the expense of incurring a high computational cost.

Various images are used to test corner detection and matching method. The images used to the corner detection are:

1) A Synthetic image

This image is used to observe how well the Harris detector handled different types of junctions [51].

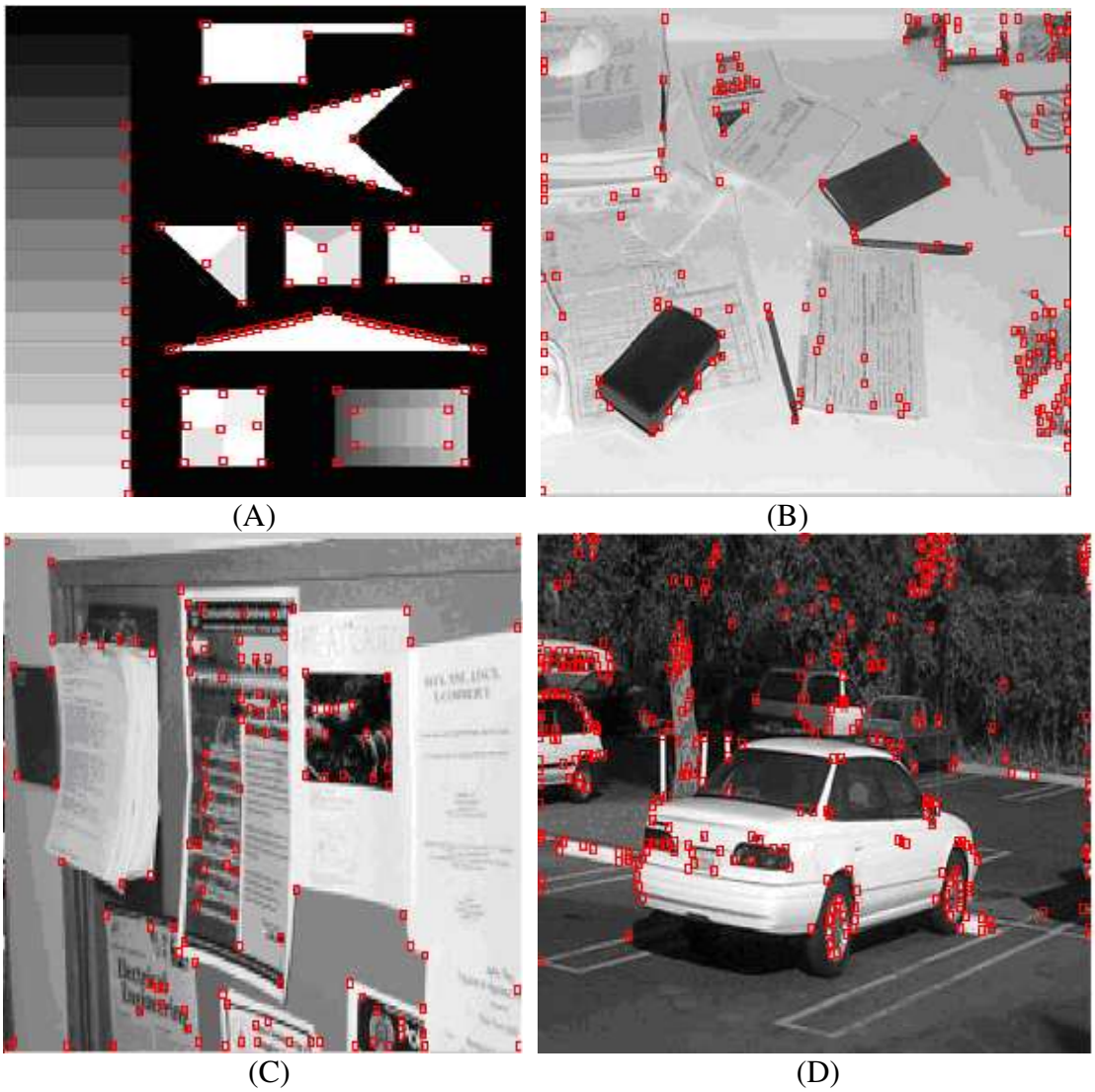
2) Indoor images

These test images are captured by a panning camera and contain a static scene with no moving objects in it. The images are used to observe how stable the corner detection is, and how the matching technique performs.

3) Outdoor images

We use some outdoor images to test how geometric corners as well as corners in textured regions can be detected. This is useful in the case of our overall approach being extended to outdoor scenes. The images contain both weakly and highly textured regions as well as geometric corners.

The Harris detector requires only one threshold: the lower the contrast in the image the lower the threshold needed, and vice-versa. A suitable range is found to be 3000 to 13000 depending on the image content. In our implementation we set the default to 4000. The amount of computation is independent of image content and $6n$ additions and $3n+10$ multiplications (where $n$ refers to the diameter of the window used) are required. For $n=5$ we get 55 operations per pixel. Figure 3.4 shows the results of the corner detection on the various images.



(A) (B)

(C) (D)

**Figure 3.4: Results of the Harris Corner detector.** (A) Synthetic image; (B) Desk image; (C) Notice board image; (D) Outdoor image.

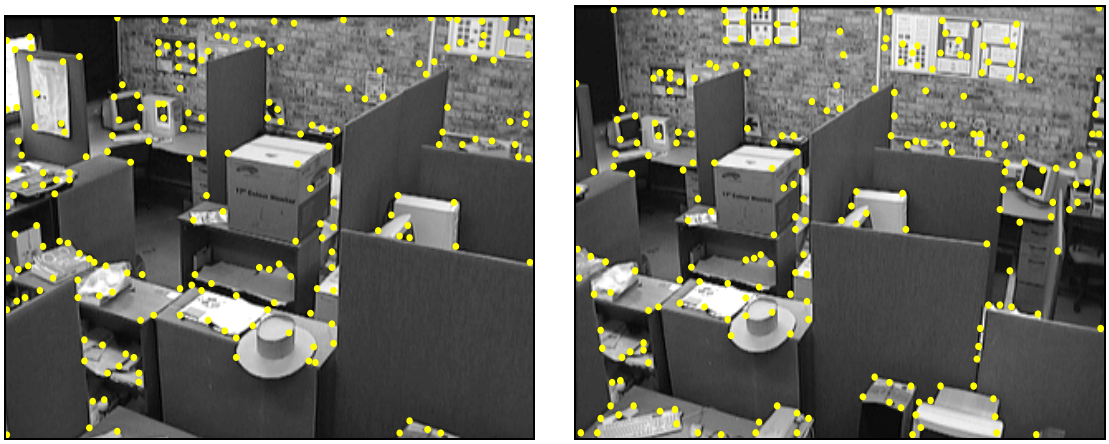
A wide Gaussian smoothing function is able to reduce noise effectively but affects the location of the detected corner. A trade-off between consistency of detection and localisation is made, especially since the Harris corner detector is known to have a lower accuracy for other types of junctions when compared to L junctions [52].

The results of the corner matching are shown next. Once again, various image types are used to test that the method conformed to the criteria set. Figure 3.5 shows two views of the Leuven castle and the results of the corner detection and matching. Figure 3.6 shows the results of our approach on a static laboratory scene captured by a panning PTZ camera. Figure 3.7 shows the results of our approach on a static laboratory scene captured again by a panning PTZ camera. Two people moving independently of the camera motion are present.

(A) (B)

(C)

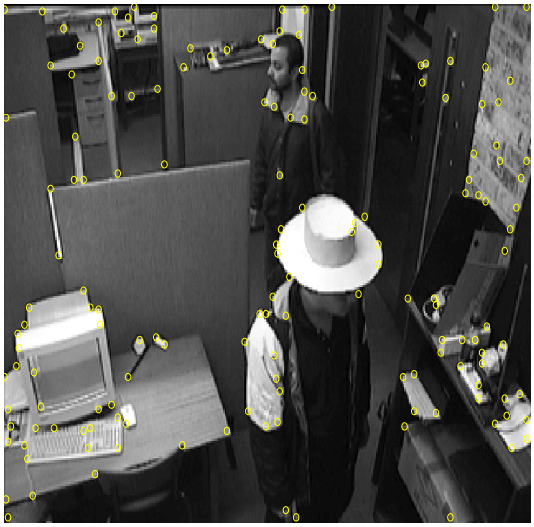
**Figure 3.5: Matching by correlation.** (A) (B) two images from the Leuven castle image sequence with the detected corners superimposed; (C) putative matches are shown by the lines linking the matching corners



(A) (B)

(C)

**Figure 3.6: Matching by correlation.** (A) (B) two images of a static laboratory scene with detected corners superimposed. The images are captured by a Pan-Tilt-Zoom camera from different viewpoints; (C) putative matches are shown by the lines linking the matching corners



(A)

(B)

(C)

**Figure 3.7: Matching by correlation.** (A) (B) two images of a static laboratory scene that are captured by a Pan-Tilt-Zoom camera and containing moving people. The detected corners are superimposed; (C) putative matches are shown by the lines linking the matching corners

# 3.5 Conclusions

In terms of stability, the detector is found to be well suited to consistently finding corners reliably enabling correspondences to be found. Several corners are detected in the images and one approach would be to restrict their number before matching them.

The number of corners detected can be reduced by using a higher threshold or only selecting corners with a specific distance between them. This could be tuned to yield the desired number of corners. A point worth noting is that most of the strong corners in the images are located in the same area and so care would have to be taken to ensure the detected corners stay well spatially distributed over the images.

The localisation error criterion is used to evaluate interest points in images and measures whether a corner is accurately located at a specific 2D location. It is, however, an intrinsic error of the corner detection and cannot be measured directly from the images [58]. From the observed results, although "conventional" corners such as L-junction, T- junctions and Y-junctions satisfied the definition of a corner as a point where the local autocorrelation of the image intensity is high, so too did locations in the image with significant texture. The Harris detector detected corners well and had a good localization performance.

Another important criterion for the use of a corner detector in our application is consistency of detection. Repeatability explicitly compares the geometric stability of the detected corners between the two images of the scene taken from different views. A corner is "repeated" if the 3D scene point in the first image is also detected in the second one. The repeatability rate is then the percentage of the total observed points in both images [41]. In our approach, corner detection is not a final result in its own right but more an input for further processing. Therefore, the true performance criterion, we feel, is how well the detection prepares itself as an input to subsequent algorithms – in our case the corner matching and homography estimation. The Harris detector is found to be computationally efficient; its computation is independent of the images and also easy to implement.

The correlation-based matching approach we use provides reasonably good matches. However, the putative matches obtained show inevitable mismatches. The matching error can be determined from the number of false matches obtained when establishing the feature correspondences in the images. Only overlapping regions in the images would be considered and a matching error determined from the true and false matches in this region. Corners that do not appear in both images would corrupt the matching

error. A mathematic formulation of determining the matching error (or repeatability rate) can be found in [40 and 41].

# Chapter 4

# Robust Transform Model Estimation and Image Warping

"I not only use the brain that I have, but all I can borrow."

- Woodrow Wilson (1856 - 1924)

*Once the features have been detected in the images, and their correspondences established, the mapping function that may be used to align the images can be estimated. As the images acquired by the camera are related by a homography, we describe the robust estimation of this transform by the RANSAC algorithm. Although there are moving objects in the scene, under the dominant motion assumption these are treated as outliers to the dominant camera motion that the homography describes.*

## 4.1 Introduction

Transform models may be divided into two broad categories depending on the amount of image data that is used as a basis for their support. A global model is composed of a single set of mapping function parameters that maps the entire image. In other words, a single equation maps each point in the one image to a corresponding location in the other image, and the parameters of this equation do not depend on the image location. Rather, all parts of the image are used to compute the mapping function parameters.

In contrast, for local models the mapping of points depends on the location of the point in the image. The local mapping functions treat the image as a composition of several smaller patches and so the parameters of the mapping function need to be defined. Only relevant local parts of the image for each set of local parameters are used in determining the local transformation. Local models are not required for the applications this thesis looks at and are not discussed further.

A homography relates any two images that are captured by a camera that rotates about its optical centre, as is the case with our PTZ camera. It is this transform we use to find the inliers from the putative matches found and subsequently align the images. A robust estimator is used to estimate the homography that is responsible for the motion of the majority of the corners, which we refer to as the *dominant* motion. Further, unless the scene is cluttered with many moving objects, this is assumed to be the motion of the camera with respect to the static background [13, 30].

## 4.2 Robust estimation of a homography

If exactly four point correspondences are given, then provided no three points are collinear an exact solution for the matrix H is possible. This is referred to as the minimal solution. These four point correspondences can manually be selected and used in generating a homography. The minimal solution is important as it defines the size of the subsets required in robust estimation algorithms. When points are measured inexactly, and if more than four such correspondences are given, then the correspondences are not totally compatible with any projective transformation. The task at hand then becomes one of determining the 'best' transformation from the data available. This is achieved by finding the transformation H that minimizes some cost function. There are two main categories of cost function: those based on minimizing an algebraic error, and those based on minimizing a geometric or statistical distance. The latter type of cost function is used in this research and is now briefly explained.

The cost function we minimize is the *Symmetric transfer error* and this function is based on the measurement of geometric distance in the images. The symmetric transfer error considers the forward (H) and backward ($H^{-1}$) transformations, and sums the geometric errors corresponding to these transformations. This error is given by

$$\sum_i d(x_i, H^{-1}x_i')^2 + d(x_i', Hx_i)^2 \qquad (4.1)$$

The first term in this sum is the transfer error in the first image and the second term the transfer error in the second image. The estimated homography is the one for which equation 4.1 is minimized.

Figure 4.1 illustrates the discussion on the symmetric transfer error.

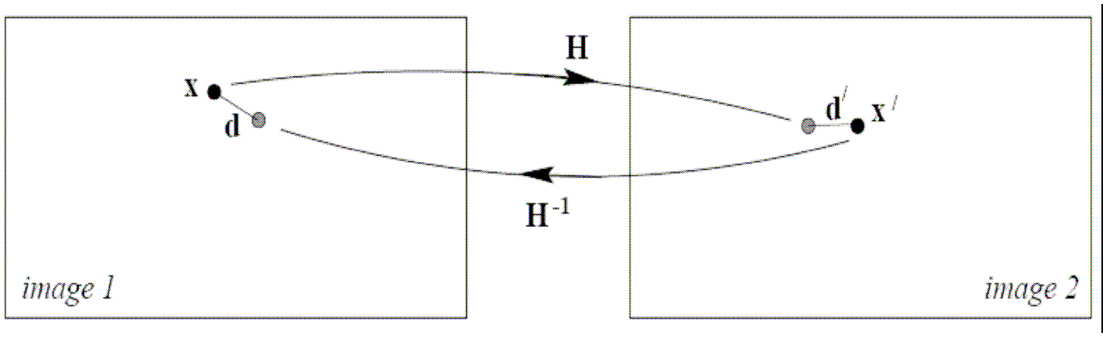



**Figure 4.1: Symmetric transfer error when estimating a homography.** The points $x$ and $x'$ are the measured (noisy) points. Using the notation $d(x, y)$ for the Euclidean distance between $x$ and $y$, the symmetric transfer error is $\sum_i d(x_i, H^{-1}x_i')^2 + d(x_i', Hx_i)^2$ .

Two more points worthy of note are mentioned next. A more detailed discussion on these points, as well as on the computation of the homography in general, can be found in [19]. The first point relates to the modelling of the measurement error or noise. In order to obtain an optimal estimate of H, it is necessary to have a model for the noise present. A common assumption is that the noise obeys a Gaussian probability distribution. This assumption not justified in general, in that it takes no account of the presence of grossly erroneous measurements (outliers) in the set of matches.

The second point concerns the selection of a coordinate system for the computation of H. In the approach we follow, points in the images are translated so that their centroid is at the origin and are then scaled so that the average distance from the origin is equal to √2. The resulting transformation is then applied to both images independently. Normalizing the data by translating and scaling the image coordinates is essential. Other than improved accuracy, a data-normalizing step allows for an algorithm invariant with respect to arbitrary choices of the scale and coordinate origin. This step undoes the effect of coordinate changes by effectively choosing a canonical coordinate frame for the measured data. After the inliers and homography have been found, de-normalization is then performed.

As is observed in the section on the feature-matching step of image registration, due to the matching being based on proximity and similarity, mismatches frequently occur in our experiments. An assumption of a Gaussian error distribution would therefore not be valid. These mismatches, which are outliers to the Gaussian distribution and can severely disturb the estimated homography, consequently need to be identified. The goal then is to determine a set of inliers from the matches so that the homography can be estimated in an effective manner.

Rigid motion imposes constraints on the motion of points between images, which the matched points between two views need to satisfy. In this thesis RANdom Sampling Consensus (RANSAC) [10] is used to automatically compute a homography between two images. The input to the RANSAC algorithm is the estimated feature correspondences in the images, and the output is the estimated homography with a set of interest points in correspondence, no other a priori information is required. Four correspondences determine a homography, so the sample size used is four. With minimal sets of four correspondences randomly selected, each set generates a putative homography. Taking more than four points is ineffective as the probability of finding a random sample of inlier matches decreases with respect to increasing sample size [19].

The support for each sample set is measured by applying the homography to all the points in the initial match set, and then counting the number of matches within a distance threshold (in our case the symmetric transfer error). For the RANSAC algorithm, a decision needs to be made regarding the number of samples and the type of sample selection taken. Firstly, degenerate samples in which three of the four points are collinear are discarded, because a homography cannot be generated from them. Next, samples that consist of points with a good spatial distribution over the images are sought. As revealed in [19], the estimated homography maps a region straddled by the computation points, but the accuracy generally deteriorates with distance from this region. This is known as the extrapolation problem and hampered our early attempts to estimate a homography. This problem is dealt with by ensuring that the images used are well textured so that the detected corners (and hence putative matches) have a good spatial distribution over both images.

To cut down the computational cost of the RANSAC algorithm, we envisage that at least one hypothesised motion will be close to the true motion we wish to obtain. If the proportion of valid data is $p(v)$, and the minimum number of features required to form a hypothesis is $m$, then the probability, $p(M_c)$ that a correct hypothesis has been encountered after $N$ iterations is approximately

$$p(M_c) = 1 - \left[1 - p(v)^m\right]^N . \qquad (4.2)$$

A stopping condition is usually determined from a desired confidence level, in our implementation, we use a probability $p(M_c) > 99\%$. Whilst $p(v)$ is generally not known in advance, a lower bound can be estimated from the largest $p(v)$ observed. An additional optimal estimation and guided matching step that can be iterated until the number of correspondences is stable is often proposed. This step is however omitted here.

To summarise, in RANSAC the support for a solution is the number of correspondences where the error is below a given threshold. The strength of this algorithm lies in the fact that it is likely to find at least one sample that only consists of inliers and thus results in a good estimate of the required homography. The robust estimator dubbed MLESAC (Maximum Likelihood Estimation Sample Consensus) [50] adopts the same sampling strategy as RANSAC in generating putative solutions to the homography and then seeking support in the remaining matches. Unlike RANSAC, which counts the number of matches that support the current homography, MLESAC evaluates the log likelihood of the solution taking into account the distribution of outliers. Errors in MLESAC are modelled as a mixture model of Gaussian and uniform distributions. Whereas in our implementation we minimise the symmetric transfer error, this algorithm is observed to minimise the *reprojection error* function [19]. As in RANSAC, it is envisaged that at least one hypothesised homography will be close to the true one if sampling and evaluation are repeated over a large number of samples. Whilst this algorithm is not used in our final implementation it shows promising improvement over the well-known RANSAC. Further details on this algorithm can be found in [49 and 50].

## 4.3 Image warping

Once the homography has been found it may be applied to one of the images to align it with the other. The task of applying a known transformation to an image is known as image warping. Transforming each pixel from the first image using the estimated homography (forward approach) would result in holes and /or overlaps in the output image due to discretization and rounding. The backward approach is therefore usually taken. First, the inverse mapping is applied to the output sampling grid, projecting it onto the input. The result is a resampling grid specifying the location at which the input is to be resampled. The input image is then sampled at these points and the values assigned to their respective output pixels. Traditionally, the Sampling Theorem defines resampling: A continuous signal may be reconstructed from its samples if the signal is band-limited and the sampling frequency exceeds the *Nyquist* rate.

The first condition avoids spectra with infinite extent that are impossible to replicate without overlap. The second condition refers to the minimum sampling frequency $f_s$. As the sampling frequency must be greater than twice the maximum frequency $f_{\max}$ present in the signal, the Nyquist frequency is the minimum distance between the spectra copies, each with bandwidth $f_{\max}$. Letting $x_k$ be a set of points located on integer positions in 2-D space. Sampling can be expressed as:

$$\hat{I}_k = \int I(x)\delta(x - x_k)dx = I(x_k) \tag{4.1}$$

If the input signal is band limited the original signal can then be perfectly reconstructed by:

$$I(x) = \sum_{k \in Z} \hat{I}_k \sin c(x_1 - x_{k1}) \sin c(x_2 - x_{k2}) \tag{4.2}$$

The interpolation is achieved by the convolution of the image with an interpolation kernel. Although sampling theory establishes the sinc function as the ideal interpolation kernel, it is not suitable for practical applications due to its infinite distribution. A number of approximations are proposed in literature, for example the nearest neighbour, bilinear and cubic spline methods of interpolation. A thorough discussion on image resampling can be found in [20, 54] to which the reader is pointed to. These references also provide the mathematical background on image warping methods. For a review of image warping methods [15] provides a good read

and excellent references on the topic. In our implementation, we use MATLAB's *imtransform* function to transform our images according to the estimated homography. We use the bilinear interpolation kernel as the default form of interpolation. The next section shows the results of the registration of two images.

## 4.4 Experimental results

The detected corners and putative correspondences that serve as the inputs to the RANSAC algorithm are found using the techniques described in Chapter 3. The default values for the number of RANSAC iterations was set to 1000 and the inlier threshold distance $t = 0.001$. As all the images used gave similar results, only two image pairs of a static scene are shown here. Figure 4.2 shows a static scene with the only movement being that due to the camera. Figure 4.3 shows a static scene containing both camera motion and an independently moving object.

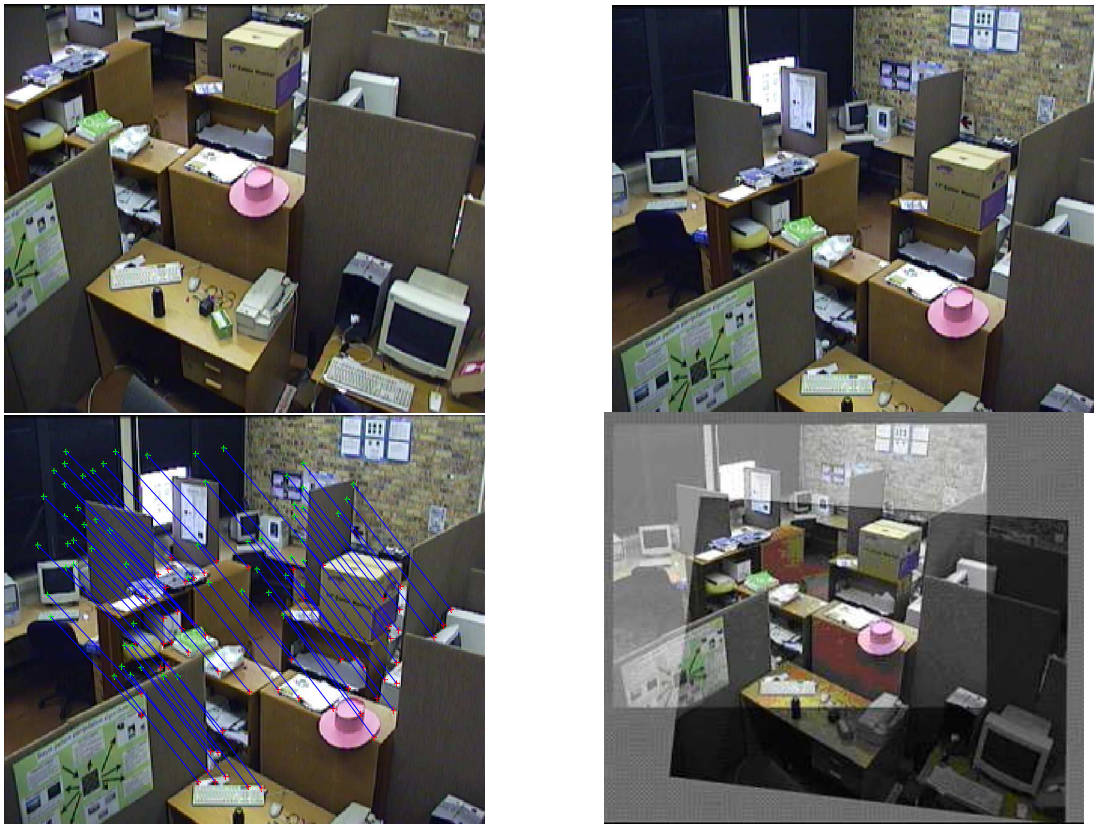


**Figure 4.2:** (Top) two views of a static laboratory scene. The images are $240 \times 320$ pixels. (Bottom) RANSAC inliers: 60 correspondences from 90 putative matches consistent with the estimated H and the aligned images.

**Figure 4.3:** (Top) two views of a static laboratory scene containing a moving object. The images are $240 \times 320$ pixels. (Bottom) RANSAC inliers: 50 correspondences from 70 putative matches consistent with the estimated H and the aligned images.

The computational efficiency of RANSAC can be improved by considering the two factors that the speed depends on: the number of samples drawn to guarantee the 99% confidence level to obtain a good estimate; and the time spent evaluating the quality of each hypothesized model (this is proportional to the size of the data set).

## 4.5 Conclusions

The estimation of the homography using the RANSAC algorithm is not affected by independent motion within the scene (provided the majority of moving pixels belong to the background) or slight changes in illumination and shadows. Despite the mismatches that were encountered, the images could still be aligned using this robust approach. However, any moving objects in the scene appear as “ghosts” in the aligned images.

# Chapter 5

# Sequence Registration and Mosaic Rendering

"Men give me credit for some genius. All the genius I have lies in this; when I have a subject in hand I study it profoundly. Day and night it is before me. My mind becomes pervaded with it. Then the effort that I have made is what people are pleased to call genius."

- Alexander Hamilton (1755 - 1804)

*In this chapter we extend the procedure used to register two images and use it to align a sequence of video frames. The creation of the resulting mosaic is achieved in two stages; sequence registration and mosaic rendering. Sequence registration estimates the point correspondences between the frames to a global model of the sequence. The rendering stage is achieved by applying a temporal operator over the registered and aligned images, resulting in a single mosaic.*

## 5.1 Global registration and sequence alignment

Global registration establishes a mapping between each frame in a sequence and an arbitrary frame. In the preceding chapters we described how to map one image to another using a homography. To extend this approach to an entire sequence a reference frame must be chosen to which the images will be warped. The choice of reference frame ultimately affects the appearance of the resulting mosaic.

If there is sufficient overlap in the frames of the sequence being considered a fixed reference frame can be chosen, and all the homographies between each image and the fixed one computed. The homographies are then used to warp each image to the fit the content of the reference frame. This is known as *frame to fixed frame* registration. In the sequences that we consider the images are registered with respect to the first

frame. However, in some applications the reference frame may not correspond to any one of the frames.

When the sequence spans a wide area the matching of features is more robust between contiguous frames. As a homography is a linear operator, the mappings between non-contiguous frames can be computed by sequentially multiplying the homographies of the in-between frames. Let $H_{\mathrm{Re}f,1}$ be the homography between the reference frame and the first image frame. The global registration is defined by the set of homographies $\{H_{\mathrm{Re}f,i} : i = 1....N\}$ where for $2 \le i \le N$

$$H_{\mathrm{Re}f,i} = H_{\mathrm{Re}f,i} \prod_{k=1}^{i=1} H_{k,k+1} \tag{5.1}$$

Once the images have been globally aligned they can be considered to form a 3-D space-time continuum as shown in figure 5.1.

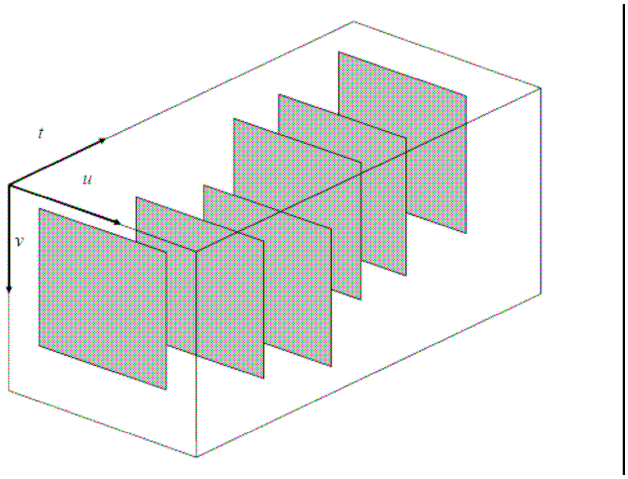



**Figure 5.1: Temporal Alignment:** in the absence of parallax, a temporal line through the image planes corresponds to the same world point.

# 5.2 Mosaic rendering

Mosaic creation is one of the applications for which image registration may be used. Having registered the frames the next step is to merge them. Several issues need to be dealt with to achieve this: the choice of reference frame or reprojection manifold onto which the images are composited; which actual frames of the sequence are to be used; and the choice temporal operator to be used for blending the images which determines how independently moving objects in the scene are handled.

Reprojection is the process of transforming every point in every image to a point in the global coordinate frame. Although the set of images and homographies form a mosaic representation of a scene, a *rendering transformation* $T$ is needed to map points in the registered images to points in the global frame. This transformation $T$ is determined by the choice of *reprojection manifold*, the surface that plays the role of the imaging sensor in the virtual camera [6]. The simplest manifold is a plane, onto which all of the images are reprojected. In this case the rendering transformation is a homography and the resulting mosaic has the classic "bow-tie form. Notably, the planar manifold cannot be used for sequences that sweep an angle larger than 90 degrees as the projective distortion means that the mosaic becomes infinite in size. Cylindrical and spherical manifolds improve the appearance of the mosaic and can handle large sweeps of up to 360 degrees. However these manifolds require a camera calibration step and therefore the simple planar projection was used in our implementation.

On regions that overlap, there are multiple contributions for the same world point on the output image. A unique intensity value that is to be used therefore has to be found. As the contributions for the same world point lie on a line parallel to the time axis the images are merged using a temporal filter. Several types of filters may be used to construct the mosaic and some of the commonly used ones are the use-last, temporal average and the temporal median filters. The use-last method uses the entire content of the most recent frame to update the mosaic. Intuitively, this visualized as placing the frames on top of each other in the order in which they are captured. Each point in the final mosaic contains the pixel value of the last frame that contributed to that point.

The averaging filter takes the average of the intensity values. The average filter is effective in removing temporal noise. However, if the sequence has moving objects on a static background these objects appear blurred in the mosaic. The median filter takes the median of the intensity values. This filter removes noise and moving objects whose intensity patterns are stationary for less than half of the frames.

One interesting form of blending images often used in computer graphics is that of *alpha blending*. Other than the three primary colour channels - red, green and blue, the fourth is known as the alpha channel. This channel conveys information about the image's transparency and specifies how foreground colours should be merged with those in the background when overlaid on top of each other. Alpha blending therefore creates the effect of transparency by combining a translucent foreground with a background colour to create an in-between blend and can be used to gradually fade one image into another.

The equation used in alpha blending is:

$$[r, g, b]_{blended} = \alpha[r, g, b]_{foreground} + (1 - \alpha)[r, g, b]_{background} \tag{5.2}$$

where [r,g,b] is the red, green, blue colour channels and alpha is the weighting factor.

The weighting factor is allowed to take any value from 0 to 1. When set to 0, the foreground is completely transparent. When it is set to 1, it becomes opaque and totally obscures the background. Any intermediate value creates a mixture of the two images or a semi-transparency. For instance, a value of $\alpha = 0.5$ would be a simple averaging of pixel values in the overlapping regions. One other form of blending that can also be achieved by using the alpha channel is the *nearest image centre*. Here, when extracting values from the input images, the distance of the sampling location from the image centre is also computed. The set of values are ranked according to this distance, and the candidate closest to its image centre is taken as the output pixel value. This is achieved using MATLAB's *bwdist* function.

Figure 5.2 on the next page illustrates the steps in forming a mosaic representation from a sequence of images and rendering a novel view. For further details on image mosaicing the reader is referred to work by Capel, Odone, Shum and Szeliski [5, 6, 31, 43, 44, and 47].

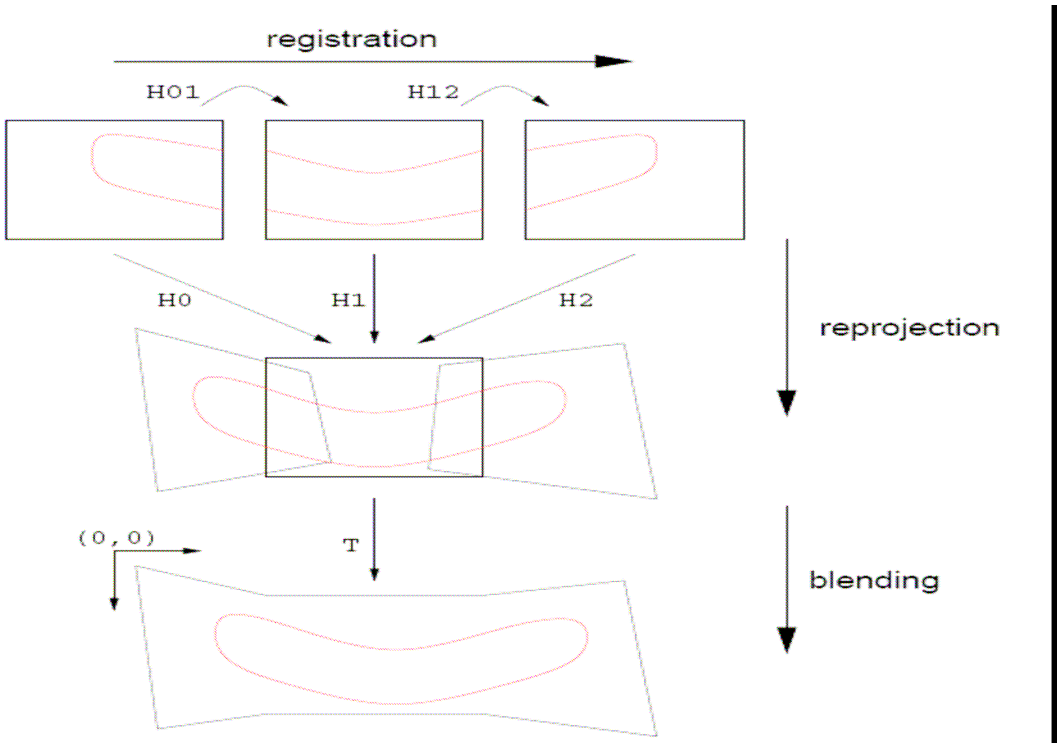



**Figure 5.2: Steps in creating an image mosaic:** here the middle frame is taken as the reference frame and the rendering transformation T shifts the origin.

## 5.3 Experimental results

The following are the results of our approach in using image registration to create mosaics. The sequences we use contain motion due to both the camera and independently moving objects. As was observed in the experiments in the previous chapter, moving objects appear as "ghosts" in aligned images. To avoid this we manipulate the value of $\alpha$ to give us our desired results. In our experiments, we were not too concerned with seams in the final mosaic but rather with how the moving objects appeared. Figure 5.3 shows the effect of alpha blending on 8 frames of two people moving in a field. The aim here was to eliminate ghosts appearing in the final mosaic. Figures 5.4 and 5.5 show how alpha blending was used to create an effect of a moving person fading out and gradually appearing, respectively, in a 16-frame sequence. This was done by changing which frame was considered as the background and which was the back ground in equation 5.2. The first 8 frames were blended together, as were the last 8. The final result was obtained by merging these two mosaics.

(a)

(b)

(c)

**Figure 5.3: Field sequence mosaic** (a) 8 frames of two people moving in a field captured by a moving camera (b) Resulting mosaic using alpha bending, $\alpha = 1$ and (c) $\alpha = 0$

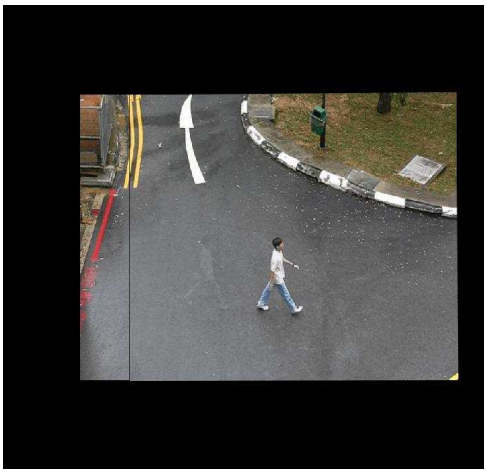
(a)

(b)

(c)

**Figure 5.4: Road sequence mosaic** (a) first 8 frames and (b) last 8 frames blended to create two mosaics. (c) The overall mosaic using alpha blending $(\alpha = 0.2)$ to give the effect of the moving person fading out.

(a)



(b)

(c)

**Figure 5.5: Road sequence mosaic** (a) first 8 frames and (b) last 8 frames blended to create two mosaics. (c) The overall mosaic using alpha blending $(\alpha = 0.2)$ to give the effect of the moving person gradually appearing.

# 5.4 Conclusions

The alpha blending does not get rid of the seams in the final mosaic. Rather, by using $\alpha = 1$ gives a mosaic equivalent to using the last frame overlaid over the first. Using $\alpha = 0$ results in a mosaic that would have been obtained using the use-last method. This results in visible seams in the final mosaic. The advantage of this method was that ghosts did not appear in the final mosaic. Alpha blending was used effectively to gradually fade out (or in) a moving person in by manipulating the appearance of ghosts the final mosaic.

# Chapter 6

# Motion detection and segmentation

"Simplicity is the ultimate sophistication."

-Leonardo da Vinci (1452 - 1519)

*In this chapter, we look at the methods of temporal differencing, thresholding and morphology. Having compensated for the camera motion by aligning images, we find the difference between the aligned images and apply a threshold to detect the independent motion. The morphological filters are used to eliminate narrow regions of falsely detected motion whilst preserving the original size and shape of the wide regions of true motion.*

## 6.1 Review of independent motion detection

Temporal differencing utilises the pixel-wise differences between two or three consecutive frames in an image sequence to detect motion. Jung and Sukhatme [24], in an approach very similar to the one we use, generate difference images whose normalised pixel values represent the probability of moving objects. The size and position of the moving objects are estimated using a Bayesian formulation based on the sequence of difference images.

Lipton et al [27] use a combination of temporal differencing and image template matching to achieve good detection, classification and tracking performance in the presence of clutter. Temporal differencing fails if the target is occluded or ceases its motion. As such they complement the temporal differencing with template matching that is most robust when the target is stationary. Moving targets are therefore detected using temporal differencing and the template-matching algorithm trained. The targets are then tracked using template matching guided by the temporal differencing stage.

In their approach, Paragios and Tziritas [36] address the detection of moving objects and their localization in two consecutive images in a sequence. They argue that the

boundaries of a moving object cannot be located precisely by inter-frame differencing alone. Instead they propose a statistical framework to model the difference image as a mixture of two zero-mean generalized Gaussian distributions, and then use a Gibbs random field for describing the label set. A maximum a posteriori criterion is used to adaptively determine the threshold for the detection of motion. The statistical framework used enables good results to be obtained even in the presence of camera motion provided this motion is estimated and compensated first. Other approaches towards achieving independent motion detection can be found in [13, 22, 26].

## 6.2 Temporal differencing and thresholding

The simplest method of temporal differencing is to take two frames and determine the absolute difference. A threshold function is then applied to determine any changes between the frames. If $I_n$ is the intensity of the *nth* frame, then the pixel wise difference function $\Delta_n$ is then

$$\Delta_n = \left| I_n - I_{n-1} \right| \tag{6.1}$$

For a stationary camera, the pixel-by-pixel subtraction method is used to detection motion as for a static scene a given 3D point continuously projects to the same position in the 2D image plane. In our implementation, a moving object is detected by finding the pixel-wise difference between two frames, having mapped pixels corresponding to the same 3D points to corresponding image plane positions (thereby compensating for the motion of the camera).

Thresholding is a well-known technique used in image segmentation that converts a multi-scale image into a binary image or *mask*. In the binary mask each pixel value is represented by a single binary digit. In its simplest form, thresholding is a point-based operation that assigns the values of 0 (black) or 1 (white) based on a comparison with a global threshold T. Thus, having found the pixel-wise difference function $\Delta_n$ a motion image $M_n$ is extracted by thresholding

$$M_n(u,v) = \begin{cases} I_n(u,v), \Delta_n(u,v) \geq T \\ 0 \quad\quad , \Delta_n(u,v) \leq T \end{cases} \tag{6.2}$$

This is known as global thresholding as a single threshold value is calculated for the whole image. Among the many global thresholding methods, is the popular and

efficient method by Otsu [33]. This method is based on an analysis of the shape properties of the grey scale level histogram of the whole image. An optimal threshold is then found according to the discriminant theory. Several other thresholding methods exist and generally thresholding methods can be grouped according to the information they exploit [39]. These categories are:

- **Histogram shape based methods** where the peaks, valleys and curvatures of the smoothed histogram are analyzed. Two major peaks and an intervening valley are searched for using tools such as the convex hull of the histogram, or its curvature and zero crossings of the wavelet component. Other authors try to approximate the histogram via two-step functions or two-pole autoregressive smoothing.
- **Clustering-based methods** in which the grey level samples are clustered in to parts as background and foreground (object) or alternatively the grey level distribution is modelled as two Gaussian distributions.
- **Entropy-based methods** result in algorithms for example, that use the entropy foreground-background regions, the cross-entropy between the original and binarized image etc. The maximization of the entropy of the thresholded image is interpreted as indicative of the maximum information transfer.
- **Object attribute-based methods** search a measure of similarity between the grey-level and binarized images, such as fuzzy similarity, shape, edges, number of objects etc. Alternatively they consider certain image attributes such as compactness or connectivity of the objects resulting from the binarization process or the coincidence of the edge fields.
- **Spatial methods** use the probability mass function models taking into account correlation between pixels on a global scale. Spatial information of object and background pixels are also utilized, for example, in the form of context probabilities, co-current probabilities, local linear dependence models of pixels, two-dimensional entropy etc.
- **Local methods** do not determine a single value of threshold but adapt the threshold value depending on the local image characteristics. These methods assume each pixel deviates according to its own model and threshold each pixel according to the context of its model. The value of the threshold depends

on some local statistics like range, variance, and surface fitting parameters or their logical combinations.

Detailed information on the various image thresholding techniques in each of these categories is found in [39] where a comprehensive survey of image thresholding methods that both describes the underlying ideas of the algorithms and measures their performances in different contexts.

For simplicity, in our implementation global thresholding is used. The global threshold T has been determined empirically to be $\approx 15\%$ of the digitizer's brightness range. As grey scale images are used, a value of $T \approx 40$ is applied to the difference image.

# 6.3 Morphological operations

Mathematical morphology is a field of non-linear image processing based on minimum and maximum operations and is used to analyse the geometric structure inherent within an image. Morphological operations allows for the systematic alteration of the geometric content of an image while preserving the stability of the important geometric characteristics. An original image is transformed into another through the interaction with another image of a certain shape and size, which is known as the *structuring element*. Geometric features in the images similar in shape and size to the structuring element are preserved whilst other features are suppressed [55]. Morphological operations therefore eliminate irrelevant objects whilst preserving the shape of larger regions.

**Definition: Translation**

Given an image $A$, the translation of set $A$ by the point $x$, denoted by $A_x$, is defined as

$$A_x = \{a + x | a \in A\}. \tag{6.3}$$

**Definition: Reflection**

Given an image $B$, the reflection of set $B$, denoted $\hat{B}$ is defined as

$$\hat{B} = \{w | w = -b, b \in B\}. \tag{6.4}$$

This operation has the same effect as rotating the image 180 degrees about its origin.

### 6.3.1 Binary erosion and dilation

The two basic building blocks for the construction of morphological operators are erosion and dilation. This section presents the underlying theory of these two operators.

**Definition: Binary Erosion**

Erosion of a binary image $A$ by a structuring element $B$, denoted by $A \Theta B$, is defined as

$$A \otimes B = \{z | z + b \in A, \forall b \in B\}. \tag{6.5}$$

The above definition of erosion can be redefined by a *Minkowski subtraction* as:

$$A \otimes B = \bigcap_{b \in B} A_{-b}. \tag{6.6}$$

where "-b" is the scalar multiple of the vector b by -1.

The erosion of the original image by the structuring element can be described intuitively by template translation. Formally, given a mask $M$ $(n \times n)$ and a part of a binary image $A$ of the same size as the mask, the erosion mask is defined as

$$A \otimes M = \begin{cases} 1 & if\ [(\forall p_{ij} \in A) \wedge (p_{ij} = 1)] \\ 0 & otherwise \end{cases} \tag{6.7}$$

Erosion shrinks the original image and eliminates narrow regions while wider ones are thinned. This is illustrated in Figure 6.1.

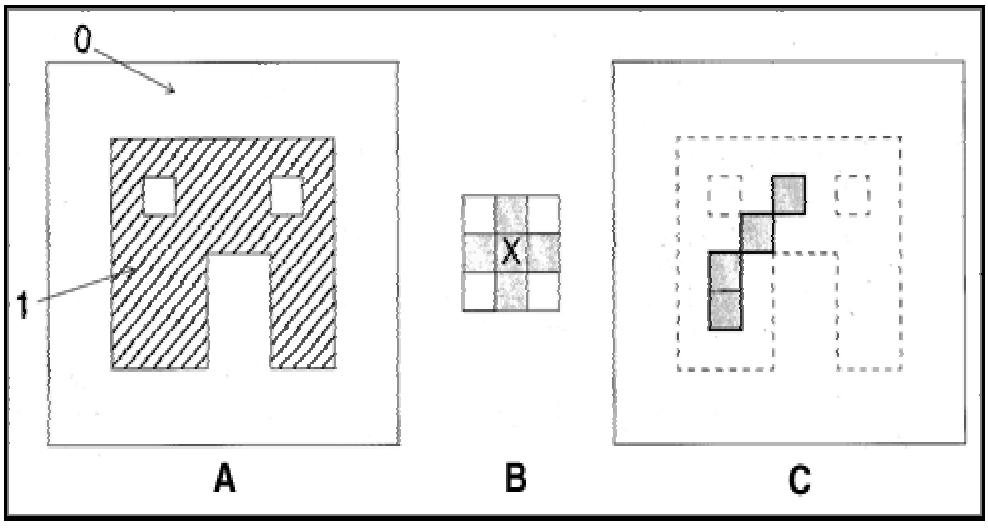



**Figure 6.1: An example of Binary Erosion.** A) Original Image, B) Structural element; x is the origin, C) Image after erosion; original in dashes.

**Definition: Binary Dilation**

With $A$ and $B$ as sets in $Z^2$, the dilation of $A$ by $B$ (where $A$ is an image and $B$ is the structuring element), denoted by $A \oplus B$, is defined

$$A \oplus B = \left\{ z \in Z^2 \middle| z = a + b, a \in A, b \in B \right\}. \tag{6.8}$$

It can be shown that the dilation is equivalent to a union of translation of the original image with respect to the structuring element:

$$A \oplus B = \bigcup_{b \in B} (A)_b . \tag{6.9}$$

Dilation is found by placing the centre of the template over each of the foreground pixels of the original image and then taking the union of all the resulting copies of the structuring element, produced using the translation. Dilation has the effect of expanding an image; so consequently, small holes inside the foreground can be filled. As with erosion, dilation can be more formally defined as

$$A \oplus M = \begin{cases} 1 & if \left[ (\exists p_{ij} \in A) \wedge (p_{ij} = 1) \right] \\ 0 & otherwise \end{cases} \tag{6.10}$$

The effect of binary dilation is shown in Figure 6.2

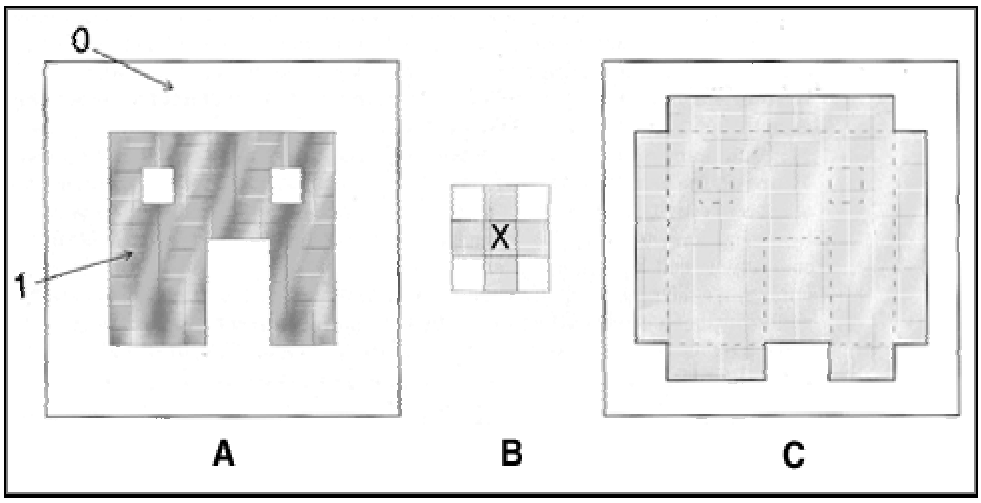



**Figure 6.2: An example of Binary Dilation.** A) Original image, B) Structural element; x is the origin, C) Image after dilation; original in dashes.

After having eroded an image to remove the narrow regions, wider regions that are thinned can therefore be restored by applying dilation with a mask of the same size. This leads us to more advanced morphological operations; morphological *opening* and *closing*.

### 6.3.2 Binary opening and closing

Binary erosion and dilation can be used in a variety of ways to give other transformations such as thinning, thickening, skeletonisation and several others. This section presents the underlying theory of two of such advanced morphological operators obtained by cascading the two basic morphological operators. These are binary *opening* and binary *closing*.

**Definition: Binary Opening**

The process of erosion followed by dilation is known as *opening*. Opening of a binary image $A$ by the structuring element $B$, is defined as

$$A \circ B = (A \otimes B) \oplus B. \tag{6.11}$$

This operation has the effect of eliminating small and thin objects, and smoothing the boundaries of larger objects without significantly changing their area. This can be thought of intuitively as "rolling the structuring element about the inside boundary of the image as is illustrated in figure 6.3.

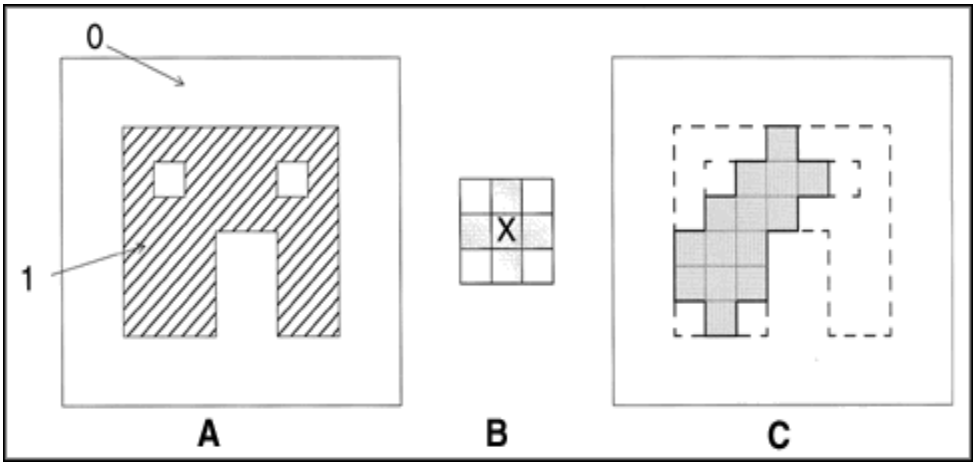



**Figure 6.3: Illustration of Binary Opening.** A) Original Image, B) Structural element; x is the origin, C) Image after opening; erosion followed by dilation.

**Definition: Binary Closing**

The process of dilation followed by erosion is called *closing*. Closing of a binary image $A$ by the structuring element $B$, is defined as

$$A \bullet B = (A \oplus B) \otimes B \tag{6.12}$$

Binary closing has the effect of filling small and thin holes in objects, and smoothing the boundaries of objects without significantly changing their area. Closing can also

be intuitively thought of as “rolling the structure element on the outer boundary of the image”.

The effect of binary closing is illustrated in Figure 6.4

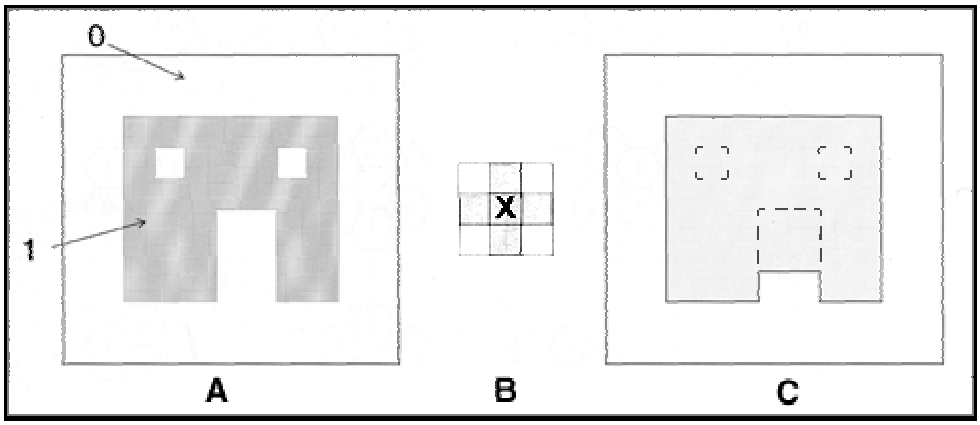



**Figure 6.4: Illustration of Binary Closing.** A) Original Image, B) Structural element; x is the origin, C) Image after closing; dilation followed by erosion; original in dashes.

Other advanced and efficient morphological operations can readily be found in most computer vision and image processing literature.

# 6.4 Experimental results

The following are the results of our approach on a short sequence captured by a non-stationary camera of a moving person. Two frames were aligned and the absolute difference between them found. As regions that do not appear in both frames need to be disregarded, these were manually cropped out. Morphological operations were then performed to remove erroneously detected regions of motion and a bounding box was displayed around the region of true motion. As real-time requirements were not an ultimate concern of this thesis, no timing results are shown. However, from the corner detection and matching stage, on average our approach took approximately 15 seconds to complete the motion detection.

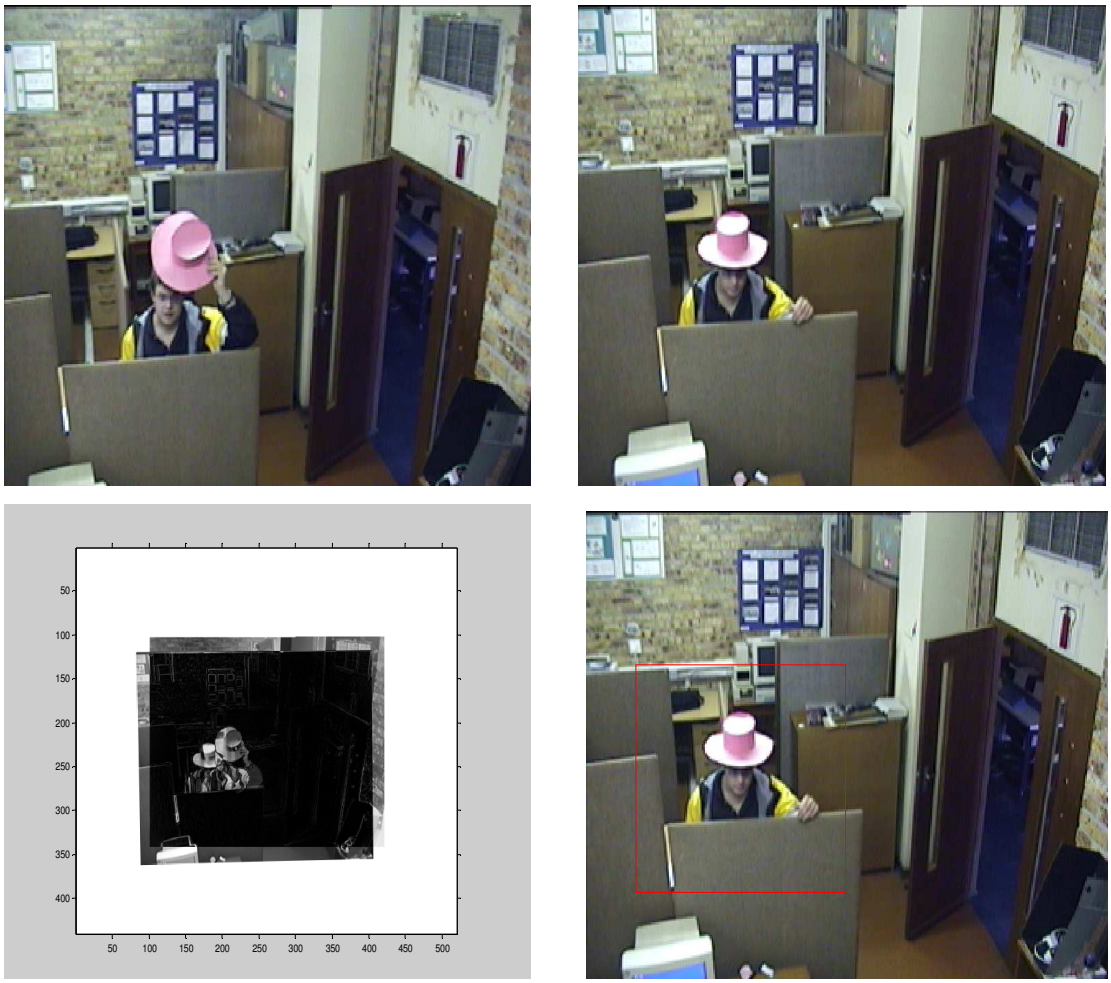



**Figure 6.5: Independent motion detection** (Top) two images of a moving person captured by a non-stationary camera. (Bottom) the frame difference between the aligned images (left) and the resulting bounding box around the detected motion (right).

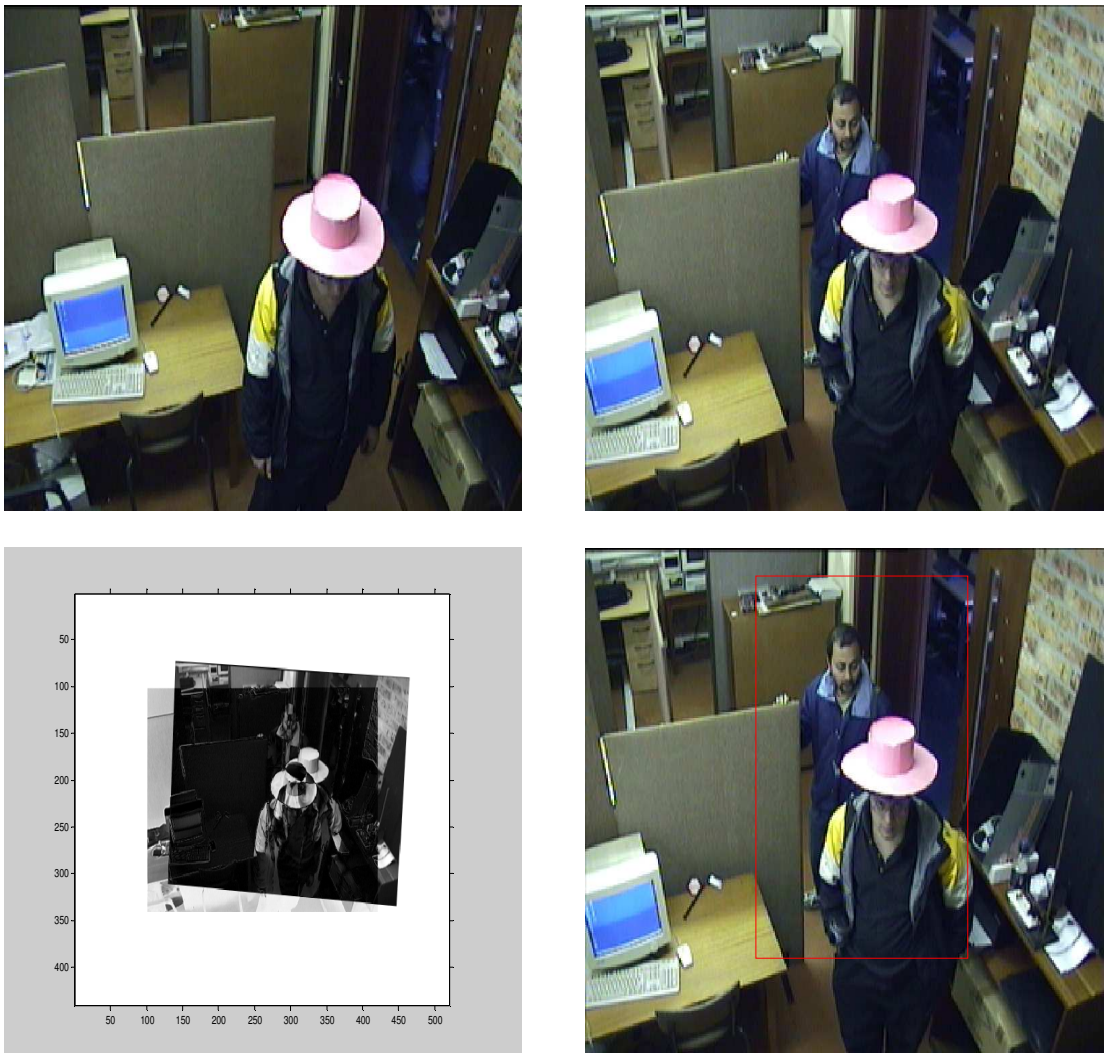



**Figure 6.6: Independent motion detection** (Top) two images of a two moving people captured by a non-stationary camera. (Bottom) the frame difference between the aligned images (left), and the resulting bounding box around the detected motion (right).

# 6.5 Conclusions

In Figure 6.6, techniques for splitting the two moving people were not explored. However, both results show that despite the camera movement regions of independent motion can be detected by using image registration. Further the approach, although not optimized for real-time operation, was fully automated and robust.

# Chapter 7

# Conclusions and Future Work

"A conclusion is simply the place where someone got tired of thinking"

-Arthur Bloch

"To raise new questions, a new possibility, to regard old problems from a new angle requires creative imagination and marks real advances in science."

- Albert Einstein (1879 - 1955)

*This thesis looked at the stages of image registration and two of its application areas. Our aim was to capture images with a PTZ camera panning a scene, register pairs of images, and subsequently investigate how the information provided could be manipulated. This chapter draws conclusions from the research and gives possible future research directions.*

## 7.1 Summary

The objectives of this thesis were to gain an in-depth understanding of image registration. We also examined two different applications of image registration, mosaicing and independent motion detection. Due to the project limitations the bulk of the work focused on the stages of image registration. The registration of image pairs was successfully extended to image sequences and mosaicing achieved. Our approach also enabled us to compensate for the camera motion (ego-motion) and apply the method of temporal differencing to detect independent motion. Applying a global threshold followed by morphological operations, erosion and dilation in cascade eliminated erroneously detected regions of motion. We now draw conclusions from each stage of our approach.

- **Corner detection and matching**

  The Harris detector worked well in detecting corners that were spatially distributed, fairly well localised and consistently detected in both images. One

pitfall though was that often too many corners were detected and this proved to be time consuming for the matching stage.

The correlation-based matching technique developed by Zhang et al [56] found good putative matches. As the matching was based on proximity and similarity, mismatches were inevitable. One factor that was not investigated was the effect of the disparity between the images on both the accuracy and speed of the matching. However, real-time detection was not the concern of this thesis.

- **Robust transform estimation and image warping**

The well-known RANSAC algorithm was found to robustly estimate the homography in the presence of mismatches. However the real-time performance of the algorithm is questionable. Further, the main assumption made was that the camera was responsible for the dominant motion. Image pairs were successfully aligned using the dominant homography that described the camera motion. Although the background was perfectly aligned, moving objects appeared as "ghosts".

- **Sequence registration and mosaic rendering**

Frames from image sequences containing moving objects were used for mosaicing. The alpha blending technique gave results equivalent to a use-first or use-last blending method depending on whether the value of $\alpha$ was 1 or 0. Although the blending was not able to produce a seamless final mosaic, it successfully removed the ghosts that would be caused by the moving objects. A value of $\alpha = 0.2$ was used to manipulating the ghosts to make the moving person fade out or gradually appear.

- **Motion detection and segmentation**

Using our approach independent motion was detected. Without aligning the images first, frame differencing would not have been able to produce meaningful results. Due to project time constraints the extension of this a whole video sequence or to a useable tracker was not investigated.

Both the mosaicing and independent motion detection via image registration worked well and were not computationally expensive. Although we use the most basic methods at each stage of our approach, we still show promising results. To incorporate the approach into a fully useable real-time tracker requires faster and

more accurate methods for each stage of our registration approach to be researched and implemented. Our approach serves as a useful foundation to achieving real-time independent motion detection and subsequent active tracking.

## 7.2 Future research directions

The applications of image registration that we presented in this thesis can be improved upon and extended in several ways. Some of the possible future research directions are now discussed.

- A faster and more accurate detection and matching scheme could be investigated. Alternatively, the number of corners detected could have some constraints added, for example a constraint on the distance between detected corners. This would reduce the time taken for the corner matching and transform estimation stages.
- The RANSAC algorithm could be optimised to be suitable for real-time purposes and a guided matching approach used to improve the matching and subsequent homography estimation.
- Median filtering could be used to remove the moving objects in the final mosaic. Alternatively a tracking algorithm could be used to identify the moving object and a mosaic of a static background created. Blending the moving object onto the static background would produce a video of the moving object in the background mosaic.
- Having used the approach outlined in this thesis to successfully detect independent motion in the presence of camera motion, the next step would be to generalise the registration approach from two images to entire sequences. Also, at present we are working with the assumption that only one independently moving object is present. Future work would include detecting and tracking several moving objects, dealing with occlusions and other such problems that were not dealt with by this thesis.
- Further the result of motion segmentation is affected by the precision of the motion estimation. Due to the inaccuracies due to noise in the frames only a rough region of the independently moving object may be obtained. As colour-segmentation can give more accurate the combination of the approach used in this thesis and the use of colour information in the images is recommended.

An ultimate goal, though beyond the scope of this research, would therefore to be to improve the approach we present and incorporate it with a colour-tracking algorithm for an efficient visual surveillance system.

# Credits

"Employ your time in improving yourself by other men's writings, so that you shall gain easily what others have laboured hard for."

-Socrates (469 BC - 400 BC)

I would like to acknowledge the following contributions to this thesis:

- The sequences that were used in chapter 5 were courtesy of Chen Bo, Marcus Chua Kok Beng and Feng Jimin of the National University of Singapore's School of Computing.
- The MATLAB functions for Computer Vision and Image Analysis provided by Peter Kovesi of the University of Western Australia. Available from: http://www.csse.uwa.edu.au/~pk/research/matlabfns/.
- The sequence used for the experiments in chapter 6 provided by Mathew Beets of the University of Cape Town's Digital Imaging Lab.
- The Leuven castle sequence courtesy of Marc Pollefeys. Available from: http://www.esat.kuleuven.ac.be/~pollefey/.
- The following websites for their amazing wealth of knowledge: http://www.robots.ox.ac.uk/~vgg and http://research.microsoft.com/~szeliski/.

The work presented in this thesis has also benefited from correspondences with various people: David Capel, Andrea Fusiello, Peter Kovesi, Marc Pollefeys, Phil Torr (who I also had the pleasure of meeting), and Miroslav Trajkovic. I am deeply indebted to you all.

# Bibliography

[1] M. Aprile, A. Colombari, A. Fusiello, and V. Murino. Segmentation and tracking of multiple objects in video sequences. In *5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2004)*, Lisboa, Portugal, 21-23 April 2004.

[2] S. T. Barnard and W. B. Thompson. Disparity Analysis of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(4), pages 333-340, 1980.

[3] P.R Baudet. Rotational invariant image operators. In *Proc. 4th Int. Joint Conf. On Pattern recognition*, pages 579-583, 1978.

[4] L.G. Brown. A survey on image registration techniques. *ACM Computing Surveys*, 24(4): 325-379, December 1992.

[5] D.P. Capel and A. Zisserman. Automated mosaicing with Super resolution zoom. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 885-891, 1998.

[6] D.P. Capel. Image mosaicing and super-resolution. PhD thesis, University of Oxford, Department of Engineering Science, Trinity term 2001.

[7] I. Cohen and G. Medioni. Detecting and tracking moving objects in video surveillance. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages II:319-325, 1999.

[8] J.E. Davis. Mosaics of scenes with moving objects. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 354-360, 1998.

[9] J.P. Donnay, M.J. Barnsley and P.A. Longley. Remote Sensing and Urban Analysis. Taylor and Francis, ISBN: 0748408606, 2001.

[10] M.A. Fischler and R.C Bolles. Random Sample Consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Comm.* ACM, 24(6), pages 381-395, 1981.

[11] A. Fusiello. Three-Dimensional Vision for Structure and Motion Estimation. PhD thesis, Università degli Studi di Trieste, November 1998.

[12] A. Fusiello. Notes on the applications of homographies in computer vision. Research Memorandum RM/99/13, Department of Computing and Electrical Engineering, Heriot-Watt University, Edinburgh, UK, 1999.

[13] A. Fusiello, M. Aprile, R. Marzotto, and V. Murino. Mosaic of a video shot with multiple moving objects. In *IEEE International Conference on Image Processing*, volume II, pages 307-310, Barcelona, Spain, September 2003.

[14] D.M Garvila. The visual analysis of human movement: a survey. *Computer Vision and Image Understanding*, 73(1), pages 82-98, 1999.

[15] C.A. Glasbey, and K.V. Mardia. A review of image warping methods. *Journal of Applied Statistics*, 25, pages 155-171, 1998.

[16] L.F. Gu. Visual guidance of robot vision. Master of Science thesis, University of Western Australia, Department of Computer Science, October 1996.

[17] M. Hansen, P. Anandan, K. Data, G. Wal, and P. Burt. Real-time scene stabilization and mosaic construction. In *proceedings of the IEEE Workshop on the Applications of Computer Vision*, 1994.

[18] C. Harris and M. Stephens. A combined corner and edge detector, Proc. *1st ECCV*, pages 118-123, 1990.

[19] R.I. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, ISBN: 0521623049, 2000.

[20] P.S. Heckbert. Fundamentals of texture mapping and image warping. MSc thesis, University of California, Berkeley, Department of Electrical Engineering and Computer Science, June 1989.

[21] A. Henrichsen. 3D reconstruction and camera calibration from 2D images. MSc thesis, University of Cape Town, Department of Electrical Engineering, December 2000.

[22] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *International Journal of Computer Vision*, 12(1): pages 5-16, February 1994.

[23] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using image stabilization. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 454-460, 1994.

[24] B. Jung and G.S. Sukhatme. Detecting moving objects using a single camera on a mobile robot in an outdoor environment. In t*he 8th Conference on Intelligent Autonomous Systems*, pages 980-987, March 2004.

[25] L. Kitchen and A. Rosenfeld. Grey-level corner detection. *Pattern Recognition Letters 1*, pages 95-102, 1982.

[26] C. Lin, C. Wang, Y. Chang and Y. Chen. Real-time object extraction and tracking with an active camera using image mosaics. Available online at http://www.cs.ccu.edu.tw/~cwlin/pub/mmspseg.pdf (last accessed 2005-05-24).

[27] A.J. Lipton, H. Fujiyoshi, and R.S. Patil. Moving target classification and tracking from real time video. *DARPA*, pages 129-136, Monterey, CA, 1998.

[28] J.B.A. Maintz and M.A. Viergever. A survey of medical image processing. *Medical Image Analysis*, 2(1), pages 1-36, 1998.

[29] H. Moravec. Towards automatic visual obstacle avoidance. *Proc. IJCAI*, page.584, 1977.

[30] F. Odone and A. Fusiello. Applications of 2D image registration. Research Memorandum RM/99/15, Department of Computing and Electrical Engineering, Heriot-Watt University, Edinburgh, UK, 1999.

[31] F. Odone. Object representation and identification in image sequences. PhD thesis, Università degli Studi di Genova, May 2002.

[32] F. Odone, A. Fusiello, and E. Trucco. Layered representation of a video shot with mosaicing. *Pattern Analysis and Applications*, 5(3), pages 296-305, August 2002.

[33] N. Otsu. A threshold selection method from gray level histograms. *IEEE Transactions on System Man and Cybernetics*, SMC-10, pages 771-774, 1980.

[34] N.R. Pal, S.K. Pal. A review on image segmentation techniques, *Pattern Recognition*, 26, pages 1277-1294, 1993.

[35] K. Paler, J. Foglein, J. Illingworth, J. Kittler. Local ordered grey levels as an aid to corner detection, *Pattern recognition* 17(5) 534-543, 1984.

[36] N. Paragios and G. Tziritas. Adaptive Detection and Localization of Moving Objects in Image Sequences. *Signal Processing: Image Communication*, pages 277-296, 1999.

[37] M. Pilu. Uncalibrated stereo correspondence by singular value decomposition. Technical Report HPL-97_96, Digital Media Department, HP Laboratories Bristol, August 1997.

[38] W.S. Rutkowski and A. Rosenfeld. A comparison of corner detection techniques for chain-coded curves, Technical report 623, Computer Science Centre, University of Maryland, 1977.

[39] B, Sankur and M. Sezgin. A Survey Over Image Thresholding Techniques And Quantitative Performance Evaluation. *Journal of Electronic Imaging*, 13(1), pages 146-165, January, 2004.

[40] C. Schmid, R. Mohr, and C Bauckhage. Comparing and evaluating interest points. In *Proceedings of the 6th International Conference on Computer Vision*, pages 230-235, January 1998.

[41] C. Schmid, R. Mohr and C. Bauckhage. Evaluation of Interest Point Detectors. In *International Journal of Computer Vision*, 37(2), 151-172, 2000.

[42] L. S. Shapiro, H. Wang, and J. M. Brady. A Matching and Tracking Strategy for Independently Moving, Non-rigid Objects. In *Proceedings of the 3rd British Machine Vision Conference*, pages 306-315, September 1992.

[43] H.-Y. Shum and R. Szeliski. Panoramic image mosaicing. Technical Report MSR-TR-97-23, Microsoft Research, September 1997.

[44] H.-Y. Shum and R. Szeliski. Construction of panoramic mosaics with global and local alignment. *International Journal of Computer Vision*, 48(2), pages 151-152, July 2002.

[45] S. Smith and M. Brady. SUSAN-a new approach to low-level image processing, DRA Technical Report TR95SMS1, 1994.

[46] P. Smith, D. Sinclair, R. Cipolla, and K. Wood. Effective corner matching. In *Proc. 9th British Machine Vision Conf.*, September 1998.

[47] R. Szeliski. Image mosaicing for tele-reality applications. In *IEEE Workshop on Applications of Computer Vision (WACV'94)*, pages 44-53, December 1994.

[48] R. Szeliski. Image alignment and stitching: A tutorial. Technical Report MSR-TR-2004-92, Microsoft Research, December 2004.

[49] B. Tordoff. Active Control of zoom for computer vision. PhD Thesis, University of Oxford Department of Engineering, Michaelmas 2003.

[50] P.H.S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. In *Journal of Computer Vision and Image Understanding*, 78(1), pages 138-156, 2000.

[51] M. Trajkovic. Motion analysis of monocular video sequences. PhD thesis, University of Sydney, Department of Electrical Engineering, March 1999.

[52] M. Trajkovic and M. Hedley. *Fast Feature Detection and Matching for Machine Vision*. In *Proc. of the 7th British Machine Vision Conference*, pages 93-102, University of Edinburgh, Scotland, September 1996.

[53] H. Wang and M. Brady. Real-time corner detection algorithm for motion estimation. *Image and Vision Computing*, 13(9), pages 695-703, 1995.

[54] G. Wolberg. Digital Image Warping. IEEE Computer Society Press, ISBN: 0818689447.

[55] H.G. Yu. Morphological image segmentation for co-aligned multiple images using watersheds transformation. MSc thesis, Florida State University, College of Engineering, Department of Electrical and Computer Engineering, Fall Semester 2004.

[56] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Loung. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial intelligence journal*, 78, pages 87-119, 1995.

[57] D. Ziou and S. Tabbone. Edge Detection Techniques: An Overview. Pattern *Recognition and Image Analysis*, 8(4), 1998.

[58] B. Zitova and J. Flusser. Image Registration Methods: a Survey. *Image and Vision Computing*, 24, pages 977-1000, 2003.