Achieving Accurate Image Registration as the Basis for Super-Resolution

Douglas Lim

This report is submitted as partial fulfilment
of the requirements for the Honours Programme of the
School of Computer Science and Software Engineering,
The University of Western Australia,
2003

Abstract

Crucial information barely visible to the human eye is often embedded in a series of low-resolution images taken of the same scene. Super-resolution enables the extraction of this information by reconstructing a single image, at a higher-resolution than is present in any of the individual images. This is particularly useful in forensic imaging, where the extraction of minute details in an image can help solve a crime.

The capturing of multiple low-resolution images taken of the same scene results in a distortion between each image. Image registration is the process of determining this distortion. This information is then used in the super-resolution process to create a set of simulated low-resolution images. The differences between these simulated images and the observed images are then used to iteratively update an initial estimate of the high-resolution image. Successful super-resolution is dependent on accurate image registration. In this thesis, we examine the hypothesis that the visual quality of a reconstructed high-resolution image improves when accurate image registration is achieved.

In the first part of this thesis, we examine the image registration process in detail. Both picture and text images are registered using two algorithms. The first registration algorithm based on an optimization approach whilst the other is based on the RANSAC algorithm. We find that the optimization approach is severely hampered by higher degree transforms such as affine transforms. This is attributed to the increased number of parameters requiring optimizing.

In the second part of this thesis, we focus on the super-resolution process. Numerous experiments were conducted to test our original hypothesis. The first experiment involved reconstructing an image when perfect registration was achieved, and comparing the results to when the RANSAC algorithm was employed. The results suggested that the visual quality of the reconstructed images were higher for perfect registration. We also found that the visual quality of reconstructed images was higher when images were registered using the RANSAC algorithm, as compared to an optimization approach.

Keywords: Image registration, super-resolution, image enhancement, image alignment, image reconstruction
CR Categories: I.4.3, I.4.5

Acknowledgements

This year has been the toughest year of my life and somehow I made it through. Helping me along the way were people that I will be forever indebted to.

First and foremost, I would like to thank my mum for her unconditional love and support. She has always been there for me pushing me on to give it my all, despite her own situation. Without her, I would not be where I am today. She is the rock in my life. I love you mum.

Secondly, thank you to my sister and brother for their constant support. I am truly blessed with a wonderful family.

Thirdly, my sincere thanks and gratitude go to my supervisor, Dr Peter Kovesi, for his constant support, understanding, and guidance throughout the year. It has been an absolute pleasure being one of your students. Thanks PK.

Finally, a special thank you to all the honours students of 2003 for your friendship and constant encouragement. I will never forget the endless moments of laughter, frustration, and happiness I shared with you all.

Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Introduction

In forensic imaging, crucial information barely visible to the human eye is often embedded in a series of low-resolution images taken of the same scene. The problem arises when extraction of this information is required [18]. However, modifying the sensor or camera to improve resolution is often prohibitive and not always possible. Likewise, simply enlarging an image causes pixelation. As a result, super-resolution is employed to estimate an image at a higher-resolution than is present in any of the individual images [3].

The capturing of multiple low-resolution images from different viewpoints or at different times, results in each image becoming distorted with respect to one another [2]. Image registration is the process of determining the optimal transformation matrix that brings these distorted images back into spatial alignment with one another. It is this registration step that is crucial in the super-resolution process. In this thesis, we restrict our attention to a transformation matrix known as a homography. A homography occurs when each low-resolution image is a view of a planar surface, or when the camera used to capture the image is rotated about an axis through the projection centre [6].

In super-resolution, the transformation matrix is used to generate a set of simulated low-resolution images [8]. The differences between this set of images and the actual observed low-resolution images, are used to iteratively update an initial estimate of the high-resolution image. Successful super-resolution is dependent on minimizing these image differences. In order to achieve this, sub-pixel accuracy in the registration process is required. Hence the need for accurate image registration [7].

Research in the area of image registration has followed several avenues. Many algorithms such as those proposed by Irani and Peleg [7], Gilles [5], and Kent [9] firstly determine an initial estimate for the transformation matrix. This initial estimate is then refined by optimizing the transformation parameters. Others

such as Stone et al. [17] determine the optimal transformation matrix by utilizing the properties of the Fast Fourier Transform of an image. Hartley and Zisserman [6] utilize the RANdom SAmple Consensus (RANSAC) algorithm of Fischler and Bolles [1] to determine the optimal homography.

Likewise, research in the area of super-resolution is varied. Irani and Peleg [8] concentrate on back-projecting differences between a set of simulated low-resolution images and the observed low-resolution images, in order to update an initial estimate image. Cohen, Avrin and Dinstein [4] build upon this work by introducing Polyphase filters, and Zomet, Rav-Acha, and Peleg [22] combine a robust median estimator in the resolution enhancement procedure.

The purpose of this research is to perform an in-depth registration accuracy analysis of current registration algorithms. Moreover, we aim to use these results to improve the visual quality of a set of low-resolution images in the super-resolution process. As a result, we propose that the visual quality of a reconstructed high-resolution image improves when accurate image registration is achieved.

CHAPTER 2

Image Registration

In this chapter, we lay the foundations for this research by discussing the image registration process, and its relation to the homography. The study then moves to 2D image transformations. The intention is to expand our understanding of transformations and find out how they are involved with the homography.

2.1 The Homography

When multiple images obtained from different sensors, different viewpoints, or at different times are taken of the same scene, they become distorted with respect to each other [2]. The problem arises when knowledge of this displacement is unknown. Image registration aims to find the optimal transformation matrix that transforms the distorted image, known as an input image, back into spatial alignment with a reference image.

In super-resolution, an initial estimate of the high-resolution image is firstly created [8]. The transformation matrix between each distorted image and the reference image is then applied to this initial estimate. This generates a set of simulated low-resolution images. The differences between this set of images and the set of actual observed low-resolution images are used to iteratively update the initial estimate. In order to achieve this, accurate image registration is vital [7].

Super-resolution is usually applied to text images. Text is usually placed on planar surfaces. As a result, we restrict our attention to a homography, which is the transformation matrix between two images of a planar surface, or between two images acquired by rotating a camera about an axis through the projection centre [6]. From now on we will use the term homography to refer to the transformation matrix.

A full homography contains eight degrees of freedom (DOF) or parameters in the transformation matrix [6]. Given that x and y represent the coordinates

of image points in the input image, and x' and y' represent the coordinates of image points in the reference image, this can be represented by

$$\begin{bmatrix} Sx' \\ Sy' \\ S \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{2.1}$$

where

- h_{11} to h_{32} represent the eight degrees of freedom that are present in a full homography, and
- S is a scale factor.

2.2 Hierarchy of 2D Image Transformations

In a 2D homogeneous coordinate system, a hierarchy of image transformations exists. These image transformations can be classified as similarity, affine, and perspective [11] as shown in Figure 2.1.

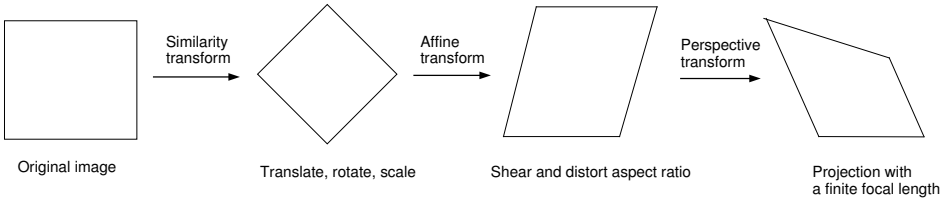


Figure 2.1: Hierarchy of image transformations in a 2D coordinate system.

A similarity transform represents the translation, rotation and scaling of an image. As a result, there are four degrees of freedom in the homography. Given that x and y represent the coordinates of image points in the input image, and x' and y' represent the coordinates of image points in the reference image, this can be represented by

$$\begin{bmatrix} Sx' \\ Sy' \\ S \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & dx \\ \sin\theta & \cos\theta & dy \\ 0 & 0 & S \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{2.2}$$

where

- S is the scaling factor,

- θ is the angle of rotation, and
- (dx, dy) is the amount of displacement or translation in the x and y directions.

An affine transform represents distortion of the aspect ratio and shearing of the image. As a result, there are six degrees of freedom in the homography. Parallel lines remain parallel and straight lines remain straight. The result of applying an affine transform to an image is given below.

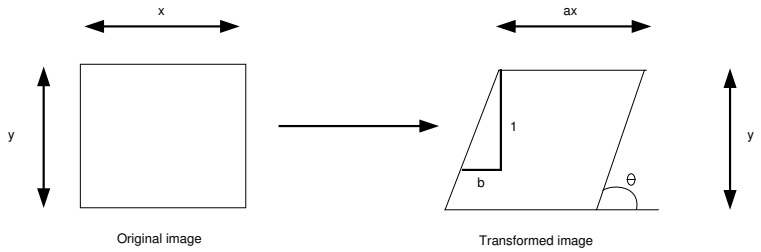


Figure 2.2: Affine transform.

The affine transform can be represented by the following homography

$$\begin{bmatrix} ax+by \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a\ b\ 0 \\ 0\ 1\ 0 \\ 0\ 0\ 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{2.3}$$

where

- a represents the scaling in x with respect to y, and
- b is the cotangent of the shearing angle θ.

Finally the perspective transform, which represents a full eight degrees of freedom in the homography, can be represented in homogeneous coordinates by

$$\begin{bmatrix} x \\ y \\ cx+dy+1 \end{bmatrix} = \begin{bmatrix} 1\ 0\ 0 \\ 0\ 1\ 0 \\ c\ d\ 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{2.4}$$

where the vanishing line of the plane is given by $[-c\ -d\ 1]^T$.

CHAPTER 3

Image Registration Algorithms

This chapter provides a detailed literature review on previous and current work in the field of image registration. Firstly, we study a general registration model based on minimizing an objective criterion. We then turn our attention to a highly successful registration model based on the RANSAC algorithm.

3.1 Minimization of an Objective Criterion

One of the most common approaches to image registration is to determine an initial estimate for the homography, and then to optimize its transformation parameters. Optimization of the initial homography usually focuses on minimizing an objective criterion.

3.1.1 Determining the Control Points

In order to determine an initial estimate for the homography, a set of control points are computed in a process known as point mapping and selection [2]. Control points are simply points that exist in the reference image that correspond to the input image. They can be classified as either intrinsic or extrinsic. Extrinsic control points are identifiable markers placed in the image specifically for registration purposes. For example, identifiable chemical markers are used in Magnetic Resonance Imaging (MRI) systems and Positron Emission Tomography (PET) to register images [2]. Intrinsic control points are determined from the image data. They are identifiable features or landmarks in the image.

Different methods of determining control points exist. Control points can be determined manually [20], where the user recognizes the positions of identifiable landmarks or features in the images by his or her eye and digitizes them with a computer mouse. However, the drawback to this approach is that the registration accuracy is limited by this initial selection of control points, and this affects

the initial estimate for the transformation parameters. If the user is relatively inaccurate in matching the control points in both the reference and input image, the initial estimate for the transformation parameters will be largely inaccurate as well.

Control points can also be determined automatically. Zheng and Chellappa [21] outline an automatic feature point extraction method based on a Gabor wavelet model and a coarse-to-fine matching technique. Ng, Larchev and Williams [13], and Hartley and Zisserman [6] use a Harris detector to automatically detect control points in their registration algorithm. The Harris detector is based on the ratio of eigenvalues of the gradient covariance matrix. Automatic selection algorithms improve registration accuracy, however they are more computationally complex and time consuming compared to the manual methods.

Other registration methods that do not use control points to identify features in an image also exist. Thevenaz, Ruttimann and Unser [19] propose a registration method that utilizes pixel intensity values across the whole image. Feature points are emphasized by some form of weighting process or binary masking of these intensity values. The advantage with this approach is that it makes use of all information available as the intensity values are spread across the image.

3.1.2 Determining the Transformation Parameters

Based on the set of matched control points, the initial estimate of the transformation parameters in the homography can now be computed. Global geometric transformations can be determined through matrix algebra, in which a matrix characterizes the transformation of the entire image [2]. Pagliari and Greene [14] represent this geometric transformation as two pixel-coordinate changing equations, which represent the mapping $(x, y) \Rightarrow (u, v)$:

$$u = \sum_{i=0}^{N} \sum_{j=0}^{N-i} a_{ij} x^i y^j,$$

$$v = \sum_{i=0}^{N} \sum_{j=0}^{N-i} b_{ij} x^i y^j,$$

where a_{ij} and b_{ij} are the unknown coefficients that require solving. In contrast, Hartley and Zisserman [6] have developed an algorithm that computes the homography between two images using a normalized direct linear transformation.

For local geometric transformations, the transformation parameters can be defined on key points and interpolated on a region-by-region basis in a process

known as elastic mapping [19]. More specifically, the "rubber sheet model" is often used [14]. This is where an image is modelled as an elastic sheet with pulling-points stretching the image. These pulling-points represent the motion of a few points in the image and is constrained to rectangular windows. Using interpolation, the motion of the remaining points in the image can be subsequently estimated.

3.1.3 Forming an Objective Criterion

The final stage in the registration process is to refine the initial estimates for the transformation parameters. This is accomplished by formulating an objective criterion, which is then minimized by optimizing the initial estimate for the transformation parameters [19]. The objective criterion measures the similarity of the input image to the reference image. Often used is the Squared Sum of intensity Differences (SSD) or variations of it, which is the mean squared intensity difference between the warped observed image and the reference image. This error measure, denoted by e, is given by

$$e = \frac{\sum_x \sum_y (I_R(x,y) - I_I(x,y))^2}{N}, \tag{3.1}$$

where

- I_R is the intensity value of the reference image at a specified pixel location;
- I_I is the intensity value of the input image at a specified pixel location;
- (x,y) is the pixel location of a point that intersects both the reference and input image; and
- N is the number of overlapping pixels between the reference image and the input image.

Both Kent [9], and Sivaramakrishna and Gordon [16] utilize the squared sum of intensity differences in their registration algorithms. The SSD is highly appropriate for uniform white Gaussian noise [9]. However it is limited when large contrast differences appear between two images. Another drawback with this approach is that it is not accurate enough to achieve sub-pixel accuracy [14].

In contrast, Thevenaz, Ruttimann and Unser [19] use an Euclidean dissimilarity measure known as the residue in their objective criterion. The residue is

simply the integrated squared difference of the intensity values. Let $Q_p f$ be a transformation parameterized by p, and q be the space dimensions. Using the notation applied in the previous equation, the residue denoted by e^2 can be written as

$$e^2 = \iint_{x\,y} (I_R(x,y) - Q_p\{I_I(x,y)\})^2 dxdy,$$

and then expressed as

$$e^2 = \iint_{x\,y} \parallel I_R(x,y) - Q_p\{I_I(x,y)\} \parallel^2 \quad . \tag{3.2}$$

Using such a criterion lends itself well to minimization with respect to p [19]. This residue measure is well known to be robust in the presence of additive white Gaussian noise. However, it lacks robustness when severe outliers in intensity values exist, such as non-stationary noise.

Irani and Peleg [7] only consider translations and rotations in their registration algorithm. As a result, they derive an error measure based on derivatives with respect to the horizontal and vertical shift, and the rotation around the origin. Irani and Peleg firstly express the horizontal shift a, vertical shift b, and rotation angle θ between two images g_1 and g_2 as

$$g_2(x,y) = g_1(xcos\theta - ysin\theta + a, ycos\theta + xsin\theta + b).$$

Next $sin\theta$ and $cos\theta$ can be expanded to the first two terms in their Taylor's series expansion, giving

$$g_2(x,y) \approx g_1(x + a - y\theta - x\theta^2/2, y + b + x\theta - y\theta^2/2).$$

The following first-order equation is now obtained by expanding g_1 to the first term of its Taylor's series expansion,

$$g_2(x,y) \approx g_1(x,y) + (a - y\theta - x\theta^2/2)\frac{\partial g_1}{\partial x} + (b + x\theta - y\theta^2/2)\frac{\partial g_1}{\partial y}.$$

Finally after a rotation θ, horizontal shift a and vertical shift b, the error measure between images g_1 and g_2 is approximated by

$$E(a,b,\theta) = \sum \left[g_1(x,y) + (a - y\theta - x\theta^2/2)\frac{\partial g_1}{\partial x} + (b + x\theta - y\theta^2/2)\frac{\partial g_1}{\partial y} - g_2(x,y) \right]^2. \tag{3.3}$$

Other optimization criteria include maximizing the normalized correlation coefficient or maximizing the mutual information of the joint pixel distributions of two images [17]. Mutual information is based on mathematical statistics [5]. Its derivation, shown below, is dependent on modelling images as random variables and determining their entropies.

For a discrete random variable, entropy is expressed as

$$H(X) = -E_X\left[\log P(X)\right] = -\sum_{x_i \in \Omega_X} \log(P(X = x_i))P(X = x_i),$$

where

- X is a random variable,
- Ω_X is the domain over which X can range,
- $P(X = x_i)$ is the probability of an event x_i, and
- $E_X[\log\ P(X)]$ denotes the expected value of $\log\ P(X)$ with respect to the random variable X.

Likewise, for a continuous random variable entropy is expressed as

$$H(X) = -\int_{-\infty}^{+\infty} p_X(x) \log(p_X(x))dx,$$

where

- $\int_a^b p_X(x)dx = P(a < X < b)$, which is the probability density of an event.

Now, given two random variables X and Y, their joint entropy is expressed as

$$H(X, Y) = -E_X\left[E_Y\left[\log(P(X, Y))\right]\right],$$

where

- P(X,Y) is the joint probability distribution of X and Y.

Finally, mutual information is now given by

$$I(X, Y) = H(X) + H(Y) - H(X, Y).$$

Relating the above equation to image registration, let X be a random variable that ranges over the domain of the model u, which represents the reference image. Therefore u(X) is a new random variable. Given that T represents the transformation that maps u onto image v, a second random variable v(T(X)) is now defined. Therefore, the mutual information between these two random variables is given by the following equation:

$$MI(T) = I(u(X), v(T(X))) = H(u(X)) + H(v(T(X))) = H(u(X), v(T(X))). \tag{3.4}$$

In order to find the optimal homography, the above equation is then differentiated with respect to the transformation T.

3.1.4 Minimizing the Objective Criterion

The final stage in the registration algorithm is to optimize the transformation parameters. This is accomplished by employing an optimization scheme to minimize the objective criterion. Optimization is the process of finding the minimum or maximum of a function $f(x_1, x_2, ..., x_n)$ of n variables, where n is an integer greater than zero [18].

Many different optimization schemes exist. Some of these methods include simulated annealing and genetic algorithms [9]. However, the disadvantage with these approaches is that they are relatively slow due to their random nature. Other techniques include Newton's method, the conjugate gradient method, and the Quasi-Newton method [18].

Kent [9] outlines a different method for minimization. He expands the error measure using Taylor's approximation, differentiates this with respect to the parameters of the transformation functions and then equates the result to zero and solves. However, the drawback to this approach is that it finds local minima rather than a global minimum. To overcome this, Kent [9] down-samples the images using a Gaussian mask and repeats the above procedure, because at lower resolutions the signal to noise ratio increases and the number of local minima decreases. The solution is then refined by repeating the above procedure at progressively higher resolutions until a threshold is reached. However, images cannot be down-sampled too much as there will be insufficient information for reliable registration.

In comparison, Thevenaz, Ruttimann and Unser [19] use a variation of an iterative gradient based algorithm known as the Levenberg-Marquardt algorithm. This is used to solve nonlinear least-squares optimization problems. By itself, the Levenberg-Marquardt algorithm is a compromise between the Gauss-Newton

method and steepest descent. However, the Levenberg-Marquardt algorithm can be computationally expensive as it calculates a Hessian matrix. To overcome this drawback, Thevenaz, Ruttimann and Unser [19] tune the algorithm to a specific application by turning optimization on or off for certain parameters. They also propose a heuristic for deciding when convergence has been reached.

3.2 Automatic Computation of Homographies using the RANSAC Algorithm

Hartley and Zisserman [6] describe an algorithm that computes a maximum likelihood estimate (MLE) of the homography based on matched interest points. Based on the RANdom SAmple Consensus (RANSAC) algorithm, developed by Fischler and Bolles [1], it may be applied to any homography.

3.2.1 Determining a Set of Putative Correspondences

The initial step of the algorithm involves determining a set of interest points in each image [6]. If each interest point in the first image can be matched to the corresponding interest point in the second image, a homography can easily be computed. Thus the problem lies in computing a homography that gives the "best fit" to the set of data.

From the set of interest points, a set of putative correspondences is determined by matching, based on proximity and similarity of intensity neighbourhoods. A square search region is centred on each interest point at (x,y) in image 1. Within this search region, the match in image 2 with the highest neighbourhood cross-correlation is selected as a putative correspondence. In some cases, one interest point in image 1 may be matched to multiple interest points in image 2. As a result, the match with the highest cross-correlation is retained. The entire process is repeated to find matches for each interest point in image 2.

Alternatively, the Squared Sum of intensity Differences (SSD) can be used as the similarity measure. Advantages include that it is computationally quicker than the cross-correlation and it is also better at detecting putative correspondences when there is a small variation in intensity between images. However, a drawback to using the SSD is that it is not invariant to the affine mapping of intensity values that often occurs between images.

3.2.2 Computing the Homography using RANSAC

Once the set of putative correspondences have been determined, the RANSAC algorithm is now employed [6]. The RANSAC algorithm is a highly successful robust estimator for fitting a model to data. For example, it may be used to determine a line of best fit to a set of 2D data points. However, as Fischler and Bolles [1] state, "The RANSAC procedure is opposite to that of conventional smoothing techniques: Rather than using as much of the data as possible to obtain an initial solution and then attempting to eliminate the invalid data points, RANSAC uses as small an initial data set as feasible and enlarges this set with consistent data when possible."

Firstly, a random sample of four putative correspondences, since four correspondences determine a homography, is selected and the homography computed. A sample should consist of points representing a good spatial distribution over the image. Also, three of the four points cannot be collinear as this leads to a degenerate case, from which the homography cannot be calculated.

The distance between every putative correspondence and the homography is then computed. The distance measure, $d_\perp$, is given by

$$d_\perp^2 = d(x, H^{-1}x')^2 + d(x', Hx)^2, \tag{3.5}$$

where

- H is the computed homography,
- $\mathrm{x} \leftrightarrow \mathrm{x}'$ is the point correspondence, and
- d(a, b) is the Euclidean distance between a and b.

However, a better distance measure that is computationally more expensive is given by the reprojection error,

$$d_\perp^2 = d(x, \hat{x})^2 + d(x', \hat{x}')^2, \tag{3.6}$$

where

- $\hat{x}' = H\hat{x}$ is the perfect correspondence.

Using this distance measure, the total number of inliers for the homography is computed. An inlier is a putative correspondence for which the distance measure $d_\perp$ is less than some threshold, which is usually set around one pixel accuracy.

All other putative correspondences that have a distance greater than the threshold are deemed to be outliers (mismatches). The homography with the largest number of inliers is taken as the best fit to the set of correspondences. If there is a tie, the homography that has the lowest standard deviation of inliers is chosen.

The entire process of sampling four random correspondences, computing the homography and choosing the one with the largest number of inliers, is repeated for multiple samples. It is infeasible and computationally expensive to repeat the process for every possible sample [6]. Consequently, the process is repeated for N samples, where N is determined adaptively.

Suppose p is the probability that at least one of the random samples of s points is free from outliers. Then if we let w be the probability that any selected point is an inlier, we have ϵ = 1 - w, the probability that any selected point is an outlier. Then we can find N by,

$$N = log(1-p)/log(1-(1-\epsilon)^s). \tag{3.7}$$

In pseudo-code, N is determined adaptively as follows:

Set N = ∞, sample_count = 0
While N > sample_count do
Choose a sample and count the number of inliers
Set ϵ = 1 - (number of inliers)/(total number of points)
Set N from ϵ and Equation 3.7 with p = 0.99
Increment sample_count by 1
Terminate

Table 3.1: Adaptive algorithm for determining the number of samples required.

3.2.3 Optimal Estimation of the Homography

Although an initial homography has been calculated using the RANSAC algorithm, it is imperative that an optimal homography is estimated [6]. In order to accomplish this, the homography is recalculated from all correspondences classified as inliers, and then optimized by using the Levenberg-Marquardt algorithm to minimize a cost function.

CHAPTER 4

Implementation of Registration Algorithms

The following chapter provides the implementation details of two widely used registration algorithms. Using MATLAB V6.5, the first algorithm implemented is based on minimizing an objective criterion. The objective criterion used is the Squared Sum of intensity Differences (SSD). The second registration algorithm implemented incorporates the RANSAC algorithm. In this chapter, we also demonstrate the registration process of each algorithm.

4.1 Minimization of an Objective Criterion

The first registration algorithm determines and optimizes an initial homography by minimizing the Squared Sum of intensity Differences (SSD). An optimization scheme is employed to find the local minimum of this error measure.

Using MATLAB's *ginput()* function, a set of control points in the reference image are digitized by the user. Likewise, the corresponding positions of these control points in each observed image is digitized.

These control points are then used to determine an initial estimate of the transformation parameters, by using the function *homography2d()* provided by Kovesi [10]. This function computes the homography between two images in a homogeneous 2D coordinate system. It is based on the normalized direct linear transformation algorithm given by Hartley and Zisserman [6]. This initial homography is then applied to the input image to bring it within spatial range of the reference image. The manual intervention used in determining an initial homography initializes the search for an optimal homography. As a result, the optimization routine is less likely to obtain local minima which may exist when the input image is spatially distanced from the reference image.

Since the input image is now within spatial alignment of the reference image,

the homography is optimized by minimizing the SSD between these two images. Given that x and y represent 2D coordinates, this error measure, denoted by e, is given by

$$e = \frac{\sum_{(x,y)}(I_R(x,y) - I_I(x,y))^2}{\#(x,y) + 1}, \tag{4.1}$$

where

- I_R is the intensity value of the reference image at a specified pixel location;
- I_I is the intensity value of the input image at a specified pixel location; and
- $\#(x,y)$ is the number of overlapping pixels between the reference image and the input image.

An integer is added to the denominator in order to ensure the error measure does not become undefined. This occurs if no overlapping pixels exist, which can be attributed to extremely poor digitization on behalf of the user. Instead, the SSD will tend to zero in the event of this occurring. This is a potential problem because the optimization routine will treat this as the global minimum. Thus when performing experiments, care was taken during the digitization process.

The optimization scheme used to minimize this objective criterion is the downhill simplex method, which is an unconstrained nonlinear optimization routine [15]. The initial estimate of the transformation parameters are passed into the optimization routine, which is employed using MATLAB's *fminsearch()* function, and are used as an initial starting point. Function evaluations are then used to determine the next lowest value until a certain tolerance is reached or the maximum number of iterations and function evaluations has been exceeded. These are set using MATLAB's *optimset()* function.

The image registration process is illustrated in Figure 4.1. The reference picture image has been translated by 10 pixels along the positive X-axis and 20 pixels along the positive Y-axis.

4.2 Computation of Homographies using the RANSAC Algorithm

As with the previously implemented algorithm, an initial estimate for the homography is firstly computed, and then subsequently refined to find an optimal

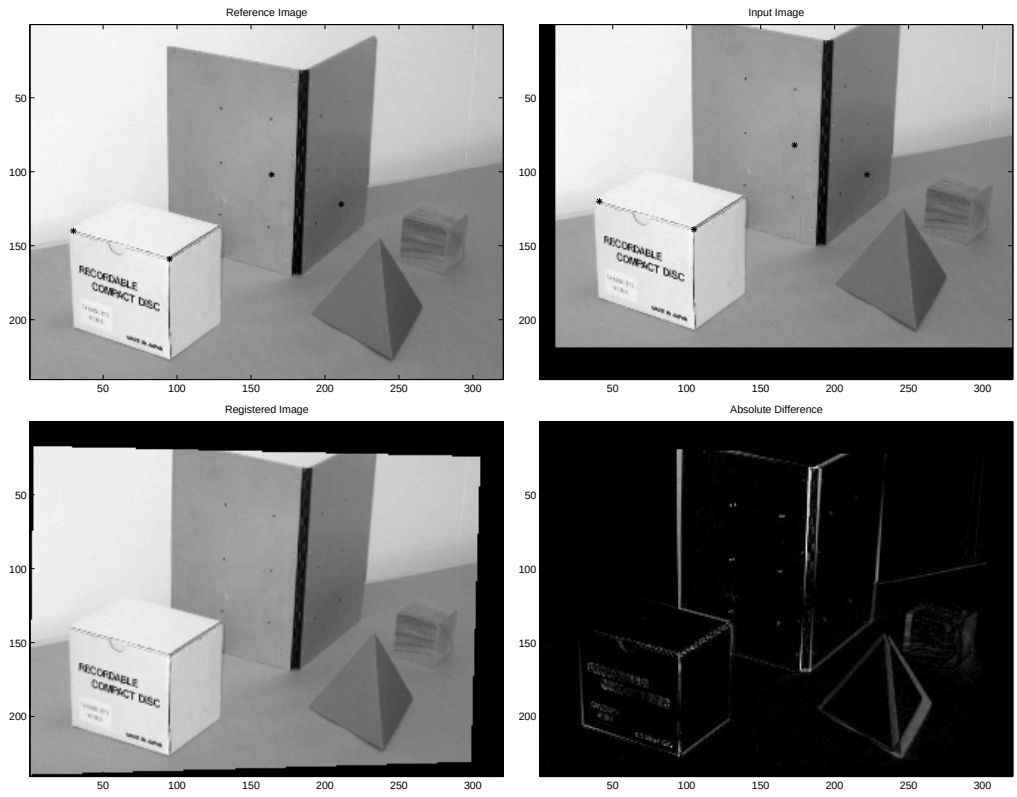


Figure 4.1: Image registration process of the minimization of the SSD. *First row of images:* Control points digitized in the reference image (left) and their corresponding locations digitized in the input image (right). *Second row of images:* Final registered image (left) and the absolute difference between the registered image and the reference image (right).

estimate. However this refinement is now provided by applying the RANSAC algorithm.

The RANSAC algorithm requires a set of putative correspondences. An initial homography is firstly applied to the input image to bring it back into spatial alignment with the reference image. Once this initial homography is applied, an implementation of the Harris corner detector, provided by Kovesi's *harris()* function [10], computes a set of feature points or interest points in both images. The number of possible matches for each feature point in the reference image, is limited by disregarding those feature points in the input image that are not within a certain pixel distance. Typically, this distance is set at 10 pixels. Next, a search window is applied around each feature point in the reference image, and those interest points within this distance in the input image. A set of putative correspondences are obtained by comparing the SSD values in these search windows, as given by Equation 4.1, and setting a certain threshold value.

Failure to apply an initial homography to the input image limits the success of the SSD measure in disambiguating putative correspondences [6]. This is particular evident when images have undergone transformations other than image translations. For transformations such as rotations or major differences in foreshortening between images, the SSD measure becomes severely degraded. A possible alternative to applying an initial homography is to use transformation invariant measures. For example, measures that are rotationally invariant may be used.

The RANSAC algorithm is applied once a whole set of putative correspondences has been determined. This RANSAC process follows the process mentioned earlier, including determining the number of random samples required as given by Table 3.1.

During the RANSAC algorithm, each random sample is checked to ensure the homography can be computed. That is, a degenerate case occurs if three out of the four points in the sample are collinear. This check is performed by forming vectors between the points and ensuring the magnitude of the cross product between the vectors will be greater than a tolerance value.

The image registration process is illustrated in Figure 4.2. The reference picture image has been translated by 10 pixels along the positive X-axis and 20 pixels along the positive Y-axis. The number of inliers found to be consistent with the refined homography is 99 out of a possible 221 putative correspondences. Note both the set of putative correspondences and inliers are shown in the figure by multiple lines linking interest points. It should be also noted that mismatches are clearly evident in the image showing the putative correspondences.

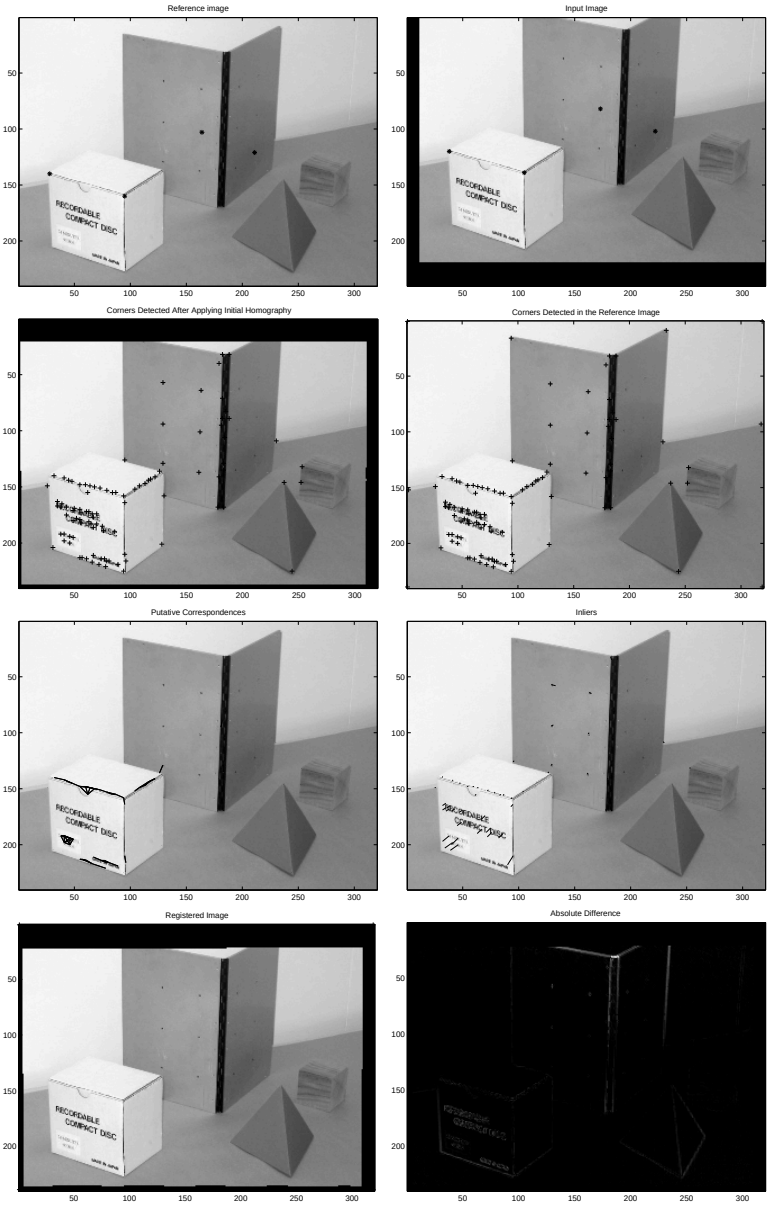


Figure 4.2: Image registration process of the RANSAC algorithm. *First row of images:* Control points digitized in the reference image (left) and their corresponding locations digitized in the input image (right). *Second row of images:* Interest points detected in the input image after the initial homography is applied (left) and the interest points detected in the reference image (right). *Third row of images:* Set of all putative correspondences superimposed on the reference image (left) and inliers superimposed on the reference image according to the estimated homography (right). *Fourth row of images:* Final registered image (left) and the absolute difference between the registered image and the reference image (right).

CHAPTER 5

Registration Accuracy

The objective of this chapter is to present an in-depth registration accuracy analysis of both implemented algorithms. We begin by firstly discussing the methodology of the experiments conducted, including test cases that will evaluate our hypotheses. We then proceed by examining the experimental results. These results will form the basis for further experimentation in the super-resolution process.

5.1 Methodology

5.1.1 Determining a Standard Measure of Registration Accuracy

Before experiments on both registration algorithms can be performed, a standard measure must be devised to record their accuracy. By devising a standard measure, different registration algorithms can be more effectively compared and analyzed regardless of implementation.

The final error measure value obtained after minimizing the SSD in Equation 4.1, is a suitable measure of registration accuracy if there are no distinct contrast differences between two images. However, it cannot be used to measure the accuracy of a registration algorithm based on the RANSAC algorithm.

Based on the homography a standard measure has been devised, as it is this matrix that governs the accuracy of a registration algorithm. If two images are perfectly registered, the pixel locations in the input image are an exact match to the corresponding pixel locations in the reference image. This knowledge is used to define the standard measure.

Firstly, a unit circle centred at the origin of a homogeneous coordinate system is generated. The homography, obtained from the registration algorithm, is then used to transform this unit circle to a new location in the coordinate system. Next the inverse of the homography is applied to transform the circle back to its

original position. If perfectly registered, the positions of each point on the circle after applying the inverse homography will be identical to the positions of the points before the homography was applied. Therefore, we define Transformation Error (TE) to be the mean magnitude of the difference between the positions of each point on the circle, after applying the inverse homography, and the positions of each point before any transformation was applied. The above process is illustrated in Figure 5.1.

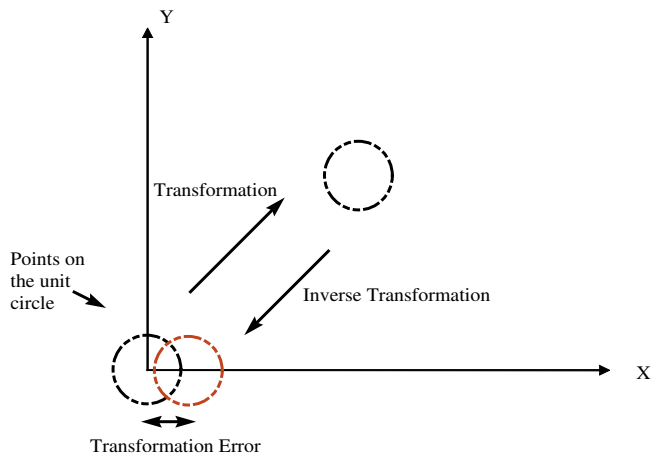


Figure 5.1: Calculation of the transformation error which is used as a standard measure to determine the accuracy of a registration algorithm.

In order to derive meaning from the transformation error with respect to the image, we must first recognize that the distance from the origin to each point on the unit circle is one. Therefore, the transformation error related to each pixel in the image will be governed by the image pixel's distance from the origin.

5.1.2 Test Cases

The first registration algorithm is dependent on an optimization routine to minimize the SSD between two images. The second registration algorithm is dependent on the RANSAC algorithm. Theoretically when transforms with a higher degree of freedom such as an affine transform exist between two images, registration accuracy is limited by the optimization routine. In addition, both algorithms rely on the SSD to either measure the difference between the input and reference images, or to obtain a set of putative matches. As mentioned previously,

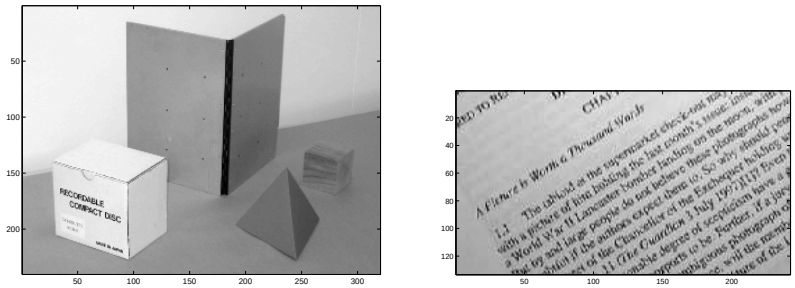
SSD is highly appropriate for uniform white Gaussian noise [9]. Therefore, we hypothesize the following statements:

1. No appreciable difference in registration accuracy exists between both algorithms for a similarity transform;

2. The RANSAC algorithm achieves greater registration accuracy for transforms with a higher degree of freedom, such as affine transforms; and

3. No appreciable difference in registration accuracy exists between both algorithms when Gaussian noise is added to the input image.

In order to evaluate the above hypotheses, a low-resolution reference image is generated by down-sampling a grey-level picture and text image. The original grey-level picture and text images are shown in Figure 5.2. Next, input images are generated according to three test cases. Finally the registration algorithm is used to register these input images with the reference image.



(a) Picture image (b) Text image

Figure 5.2: Low-resolution reference images that are used in the registration process.

The first test case involves applying a combination of small translations in the x and y directions, and rotations in a clockwise direction around the origin to the reference image. The maximum shift in the x and y directions is restricted to five pixels while the maximum rotation around the origin is restricted to one radian. Each shift in the x and y direction is incremented by one pixel, while each rotational shift is incremented by 0.2 radians. Large translations and rotations did not have to be considered due to the manual use of control points. These control points determine an initial homography that brings input images within spatial alignment of the reference image.

The second test case involves applying different affine transforms to the reference image. The maximum scaling in x with respect to y, denoted by a, is

restricted to 1, while the maximum cotangent of the shearing angle θ, denoted by b, is restricted to 0.4.

The third test case involves applying Gaussian noise to the input images, which have been generated according to the first and second test cases. However, Gaussian noise has only been applied to the picture image since it degrades the text image. As a result, it is extremely difficult to digitize control points properly. Applying Gaussian noise to the input images allows us to evaluate the accuracy of each registration algorithm in the presence of noise.

In all test cases, the optimization routine for the first registration algorithm is set to terminate at either 100 iterations, 100 function evaluations or once a tolerance value of 0.001 has been reached.

5.1.3 Determining the Interpolation Scheme

To generate the input images in the first test case, a function is created that shifts the image along the X-axis and Y-axis, and rotates the image around the origin in a clockwise direction. Likewise to generate the input images in the second test case, a function is created that applies a specified homography to the image. This specified homography will contain altered values of a and b, from Equation 2.3, that cause shearing of the image.

However, when translating, rotating or shearing an image, the resultant image will contain new pixel locations not present in the original image [12]. As a result, interpolation is required to estimate the intensity values of these new pixel locations.

MATLAB provides three different interpolation methods [12]. Each of these methods work fundamentally in a similar way. For each method, the interpolation value for an output pixel is determined by a two step process. The first step is to find the location in the input image that corresponds to this output pixel. Next, the output pixel is assigned a value by computing a weighted average of the distance between a set of pixels in the vicinity of that location and the output pixel itself.

The three interpolation methods supported by MATLAB are:

- Nearest neighbour interpolation, where the output pixel value is assigned the value of the nearest pixel in the input image;
- Bilinear interpolation, where the output pixel value is assigned the value of a weighted average of pixels in the nearest 2-by-2 neighbourhood in the input image; and

- Bicubic interpolation, where the output pixel value is assigned the value of a weighted average of pixels in the nearest 4-by-4 neighbourhood in the input image.

The first registration algorithm computes an optimal homography by minimizing the SSD between the reference and input images. Likewise, the second registration algorithm utilizes the SSD to determine a set of putative correspondences. Therefore the choice of interpolation, when estimating intensity values in the output image, will affect the success of the registration algorithm in determining this optimal transformation. As a result, we must analyze the behaviour of the SSD between the reference and input images for different interpolation schemes to determine the most effective scheme.

Figure 5.3 contains plots of the SSD value over a range of translations along the X-axis, using different interpolation schemes. Each plot is obtained by shifting the images, shown in Figure 5.2, along the X-axis from -20 to 20 pixels in increments of 0.1, and then plotting the SSD value between the shifted image and the original image.

The use of a nearest-neighbour interpolation scheme produces a plot that contains a step-like graph descending towards the global minimum at zero, which signifies a SSD value of zero when the image has not been shifted. These steps result from the output pixels assuming the value of its nearest neighbour in the input image. Therefore, pixels that surround a particular pixel will have the same values, resulting in identical SSD values for a range of small shifts. Due to its non-smooth, step-like behaviour, use of the nearest-neighbour interpolation scheme results in the optimization routine being computationally expensive in determining the optimal transformation.

The bilinear and bicubic interpolation process seems to introduce artifacts that result in many local minima in the error displacement curves shown in Figure 5.3. For the text image, this is further amplified by the production of extra local minima. These extra local minima result from the optimization routine not being able to differentiate the correct line of text when a small pixel shift has been applied. This difficulty is due to the regular structure and repeated pattern of text. As a result, this will create problems for registration algorithms based on optimization. However, we eliminate this problem by incorporating manual control points in the registration process. This brings the input image within spatial range of the reference image.

Both schemes produce a smoother curve than nearest-neighbour interpolation. This is due to the weighted average of pixels in the nearest neighbourhood being used to determine interpolated values. The SSD error measure plot of the bicubic

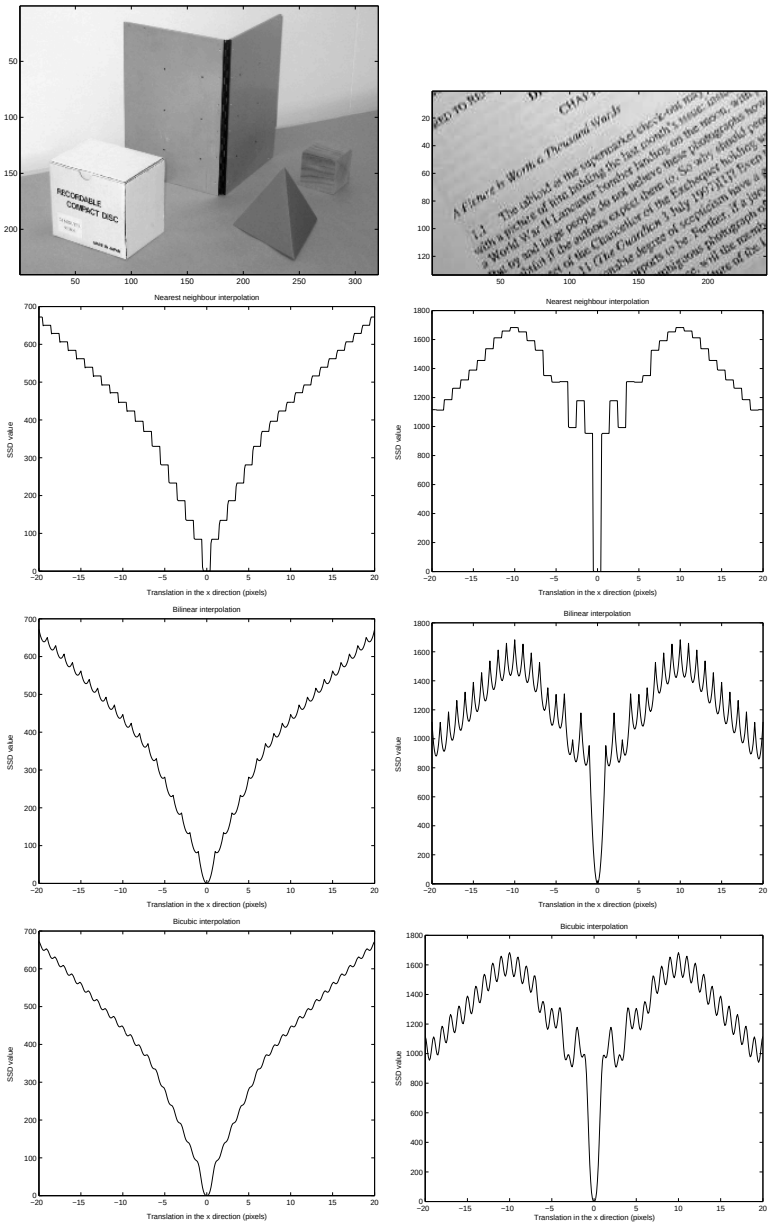


Figure 5.3: SSD error measure plots for different interpolation schemes. *First row of images:* Picture image (left) and text image (right). *Second row of images:* Nearest neighbour interpolation for the picture image (left) and the text image (right). *Third row of images:* Bilinear interpolation for the picture image (left) and the text image (right). *Fourth row of images:* Bicubic interpolation for the picture image (left) and the text image (right).

interpolation technique contains less local minima compared to the bilinear plot. As a result, the likelihood of the optimization routine obtaining a sub-optimal solution due to falling into false global minima instead of a global minimum, is reduced. Therefore, a bicubic interpolation scheme is chosen when applying a similarity and affine transform to images.

5.2 Results

5.2.1 Registering Picture Images

The results from the first test case confirm that no appreciable difference in registration accuracy exists between both algorithms for a similarity transform. Note that these results, along with the other two test cases, can be found in Appendix A. Shown below is a plot of the transformation error associated with each algorithm, for different similarity transforms. The trial number corresponds to each combination of shifts and rotations. That is, trials 1 to 5 refer to when the x shift is kept constant at one pixel, while the y shift increases from one pixel to five pixels, and at the same time the angle of rotation increases from 0.2 radians to 1 radian. Likewise trials 6 to 10 is an exact copy of trials 1 to 5 except that now the x shift is kept constant at two pixels. This process is repeated until the x shift is set at a constant value of five pixels.

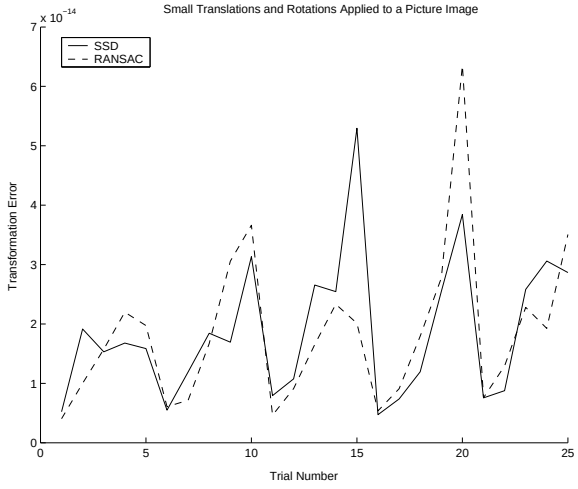


Figure 5.4: Plot of the transformation error associated with each algorithm for different similarity transforms.

From the above graph, no clear indication exists as to which registration

algorithm is more accurate in the registration process. This is especially true since the optimization routine is able to optimize three degrees of freedom.

Likewise, the results from the second test case confirm that the RANSAC algorithm achieves greater registration accuracy for transforms with a higher degree of freedom. Shown below is a plot of the transformation error associated with each algorithm, for different affine transforms. In this case, trials 1 to 4 refer to when a, the scaling in x with respect to y, is kept constant at 0.7, while b, the cotangent of the shearing angle θ, increases from 0.1 to 0.4. Likewise trials 5 to 8 is an exact copy of trials 1 to 4 except that now a is kept constant at 0.8. This process is repeated until a is set at a constant value of 1.0.

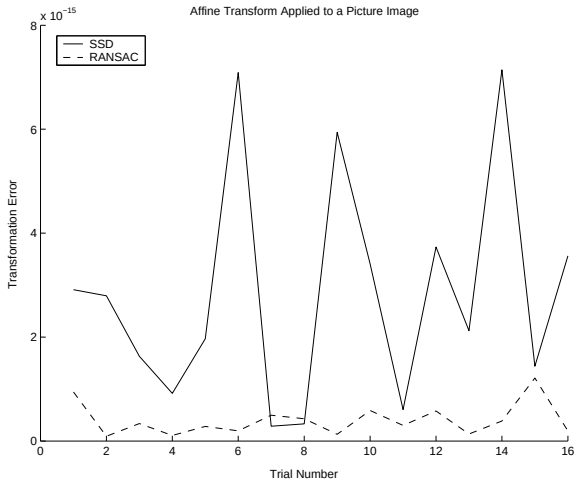


Figure 5.5: Plot of the transformation error associated with each algorithm for different affine transforms.

From the above plot, it is evident that the RANSAC algorithm achieves greater accuracy when registering picture images that have undergone an affine transform. The algorithm based on an optimization routine has to now handle a higher degree of freedom. As a result, optimization results suffer when a greater number of parameters are introduced.

Finally, the results from the third test case confirm that no appreciable difference in registration accuracy exists between both algorithms, when Gaussian noise is added to the input image. Shown in Figure 5.6 and Figure 5.7 are two plots. The first plot contains the results of the registration process when the exact same similarity transforms incorporated in the first test case, are applied to the picture image. However, Gaussian noise is now added to this image. Likewise, the second plot contains the results when the affine transforms, incorporated in the second test case, and Gaussian noise are applied to the picture image.

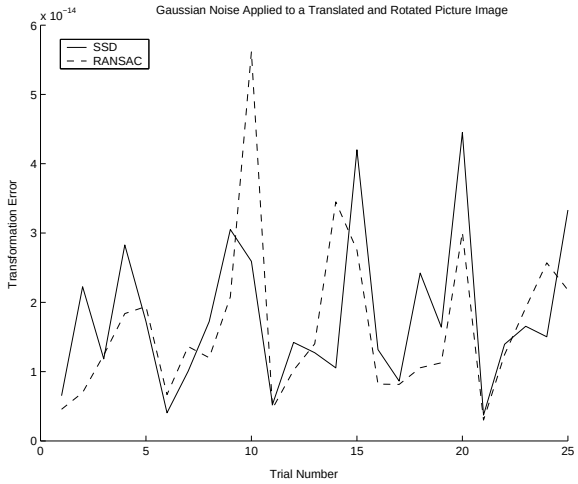


Figure 5.6: Plot of the transformation error associated with each algorithm for different similarity transforms. Gaussian noise has now been added to the input image.

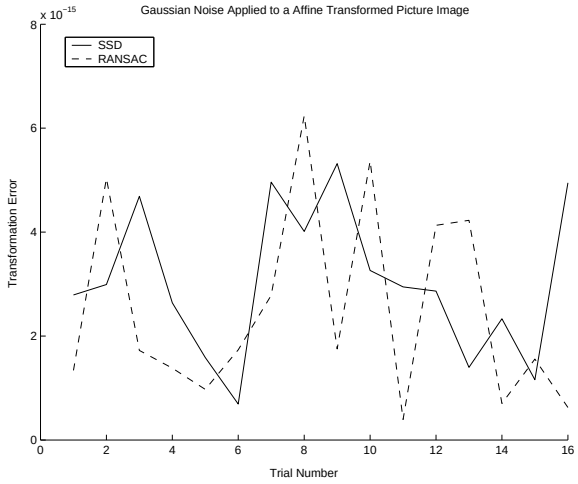


Figure 5.7: Plot of the transformation error associated with each algorithm for different affine transforms. Gaussian noise has now been added to the input image.

In both graphs there is no appreciable difference in registration accuracy between the two algorithms, confirming our earlier hypothesis. Since both algorithms rely on the SSD measure to either optimize the homography or to determine a set of putative matches, Gaussian noise added to the input images will return the same result.

5.2.2 Registering Text Images

Similar results are obtained when registering text images. These results can be found in Appendix A. Figure 5.8 and Figure 5.9 contain plots of the transformation error associated with each algorithm, after different similarity and affine transforms are applied to the reference image respectively. In this case, the reference image is now a text image as compared to a picture image. These results confirm our earlier hypotheses.

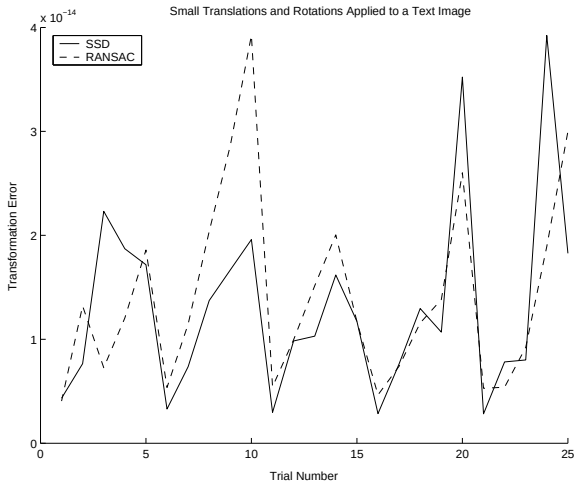


Figure 5.8: Plot of the transformation error associated with each algorithm for different similarity transforms.

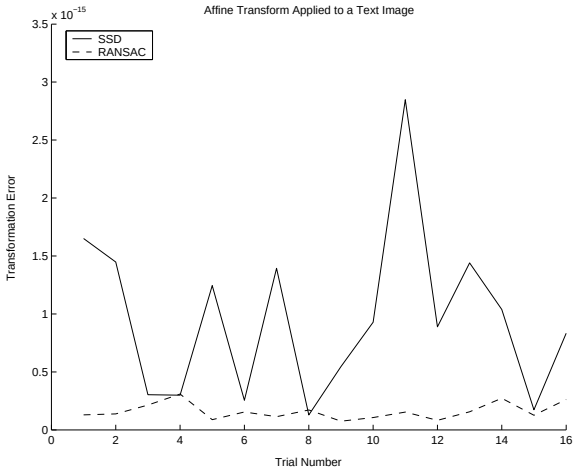


Figure 5.9: Plot of the transformation error associated with each algorithm for different affine transforms.

CHAPTER 6

Super-Resolution

Thus far we have studied the image registration process in detail. Image registration is the basis step for super-resolution. Therefore, we now focus on the super-resolution process.

6.1 The High-Resolution Image

Super-resolution is the process of enhancing the visual quality of a sequence of observed low-resolution images by reconstructing a single high-resolution image [8]. In order to successfully reconstruct a high-resolution image, the observed images need to be registered so that each pixel in each low resolution image can be associated with the correct region in the original scene. Hence the need for image registration [22].

The spatial resolution of each observed low-resolution image is influenced by the sensor's physical characteristics such as its optics, its density of the detector elements and the spatial response of the detector elements [8]. Modifying the sensor or camera to improve resolution is often prohibitive and not always possible. In addition, simply enlarging the image causes pixelation. Therefore, post-processing via super-resolution is required to restore the degraded image [18].

Super-resolution is not applied routinely for every image requiring enhancement. In fact, it is a hand-crafted approach that requires individual parameter tuning for each image. One automated solution for all types of images is not possible.

CHAPTER 7

Super-Resolution Algorithms

In this chapter, we present a detailed literature review on previous and current work in the field of super-resolution. Note that many current super-resolution algorithms are based on the work by Irani and Peleg [7, 8].

7.1 Irani and Peleg's Approach

The approach by Irani and Peleg [8] in reconstructing a high-resolution image handles dynamic images of an object, and more complex motions than pure translational motion in the image plane. Their algorithm is based on generating a set of simulated low-resolution images. The image differences between this set of images and the actual observed low-resolution images are back-projected, using a back-projecting kernel, onto an initial estimate of the high-resolution image. Figure 7.1, adapted from Irani and Peleg [8], illustrates the super-resolution process.

The generation of each observed image is the result of simulating an imaging process, which is the process where the observed low-resolution images are obtained from the high-resolution image. The imaging process can be modelled by the following equation:

$$g_k(m,n) = \alpha_k(h(T_k(f(x,y))) + \eta_k(x,y)), \tag{7.1}$$

where

- g_k is the k^{th} observed image,
- f is the high-resolution image that the algorithm is trying to find,
- T_k is the 2D transformation that maps f to g_k,
- h is a blurring function that is dependent on the Point Spread Function (PSF) of the sensor,

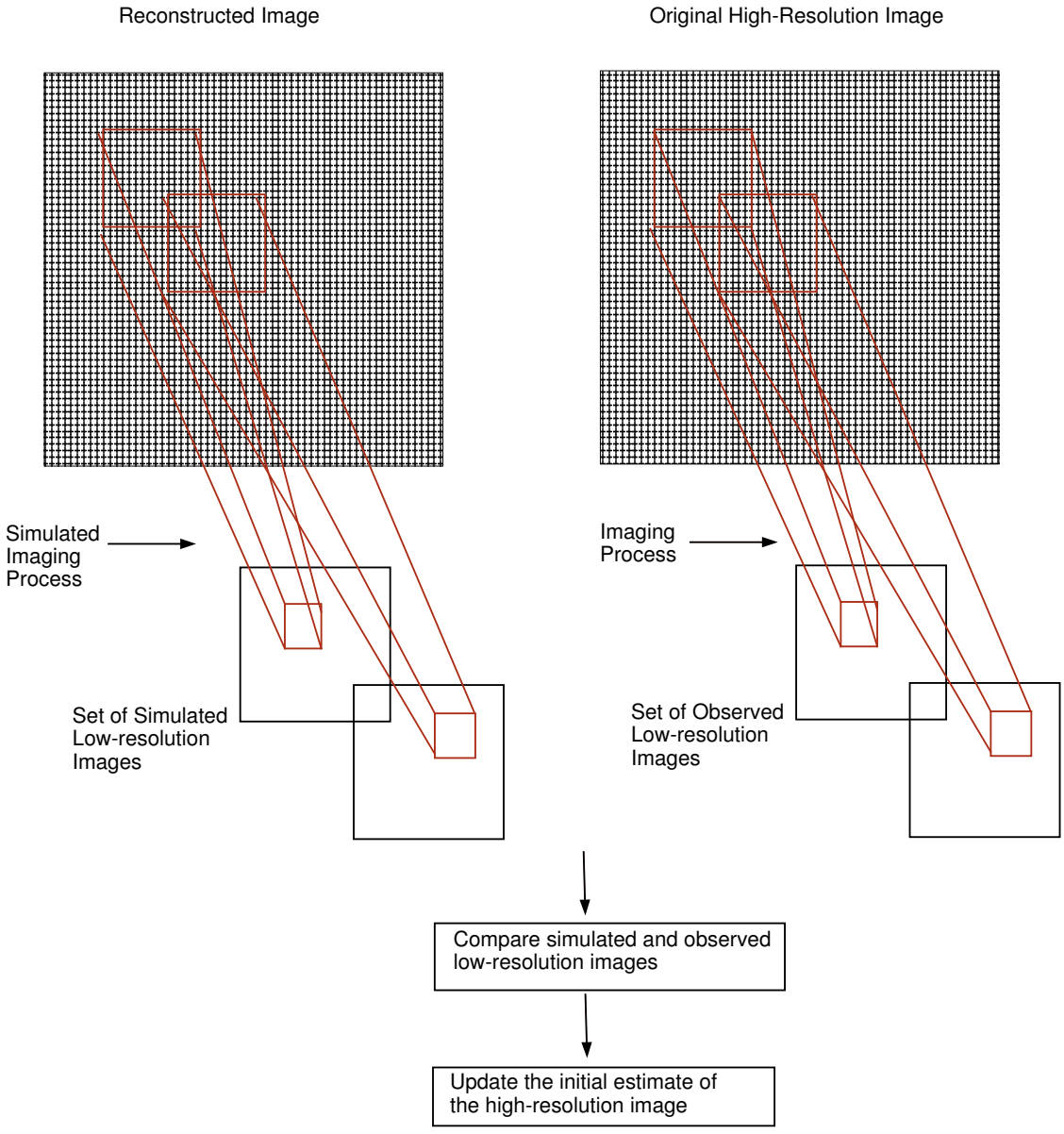


Figure 7.1: The super-resolution process proposed by Irani and Peleg. The initial estimate of the high-resolution image is iteratively updated so that the simulated low-resolution images are as close as possible to the observed low-resolution images.

- η_k is an additive noise term, and
- α_k is a down-sampling operator.

The initial stages of the super-resolution algorithm involve creating an initial estimate $f^{(0)}$ of the high resolution image, and then simulating a set of low-resolution images. This set of low-resolution images $\{g_k^{(0)}\}_{k=1}^K$ correspond to the set of observed images $\{g_k\}_{k=1}^K$. The process that yields these simulated low-resolution images can be expressed by the following equation:

$$g_k^{(n)} = (T_k(f^{(n)}) * h) \downarrow s, \tag{7.2}$$

where

- $\downarrow s$ is a down-sampling operation according to a scale factor s,
- n is the n^{th} iteration, and
- * is the convolution operator.

The differences between each simulated image and its corresponding observed image are now used to update the initial estimate image. If the initial estimate image $f^{(0)}$ is the correct high-resolution image, then the set of simulated low-resolution images $\{g_k^{(0)}\}_{k=1}^K$ should be identical to the set of observed low-resolution images $\{g_k\}_{k=1}^K$. Therefore, these image differences $\{g_k - g_k^{(0)}\}_{k=1}^K$ can be used to improve the initial guess image $f^{(0)}$ in order to obtain a high-resolution image $f^{(1)}$. Each value in the difference images is back-projected onto its receptive field in the initial guess image $f^{(0)}$.

The above process is repeated iteratively in order to minimize the following error function:

$$e^{(n)} = \sqrt{\frac{1}{K}\sum_{k=1}^{K} \parallel g_k - g_k^{(n)} \parallel_2^2}. \tag{7.3}$$

The iterative update scheme for the super-resolution process can now be expressed as follows:

$$f^{(n+1)} = f^{(n)} + \frac{1}{K}\sum_{k=1}^{K} T_k^{-1}((g_k - g_k^{(n)}) \uparrow s) * p), \tag{7.4}$$

where

- K is the number of low-resolution images,

- $\uparrow s$ is an up-sampling operation according to a scale factor s,
- p is the back-projection kernel used to deblur the image, and
- * is the convolution operator.

7.2 Cohen, Avrin and Dinstein's Approach

Cohen, Avrin and Dinstein [4] have extended the work by Irani and Peleg [8] in reconstructing a high-resolution image. Based on Irani and Peleg's work, resolution enhancement is based on the assumption that every low-resolution pixel is a 'projection' of its receptive field in the high-resolution image. However, Cohen, Avrin and Dinstein's approach to super-resolution is restricted to input images that are simple translations of the original image. Like Irani and Peleg, an initial guess for the high-resolution image is firstly generated. This initial guess image is iteratively updated by back-projecting the differences between the simulated and observed images onto its receptive field in the initial guess image. However, Cohen, Avrin and Dinstein introduce the use of Polyphase filters in the simulation of observed low-resolution images and for the back-projection of the image differences [4].

Cohen, Avrin and Dinstein assume that the observed images have only undergone translations. Let f^n denote the n^{th} image of the super-resolution sequence. Then during the super-resolution process, an image $\vec{f_n}$ is formed by modifying the coordinates of f^{n-1} such that the translation parameters (dx,dy) fulfill dx $\epsilon[0,1)$ and dy $\epsilon[0,1)$. The translation parameters are then quantized to take values of dx $\epsilon[0,1/Q,...,(Q-1)/Q]$ and dy $\epsilon[0,1/Q,...,(Q-1)/Q]$, giving Q^2 different quantization levels. For the following equations, let i and j denote the index of the quantized level of dx and dy, respectively.

For ease of convenience, let k denote the image index, h^{PSF} denote the blurring function and h^{BP} denote the back-projection kernel. Therefore, we denote the polyphase decompositions of h^{PSF} and h^{BP} by

$$h_{ij}^{PP\_PSF}(x,y) = h^{PSF}(xQ+i, yQ+j)\ i,j = 0,1,...,Q-1, \tag{7.5}$$

$$h_{ij}^{PP\_BP}(x,y) = h^{BP}(xQ+i, yQ+j)\ i,j = 0,1,...,Q-1. \tag{7.6}$$

The algorithm firstly starts by generating an initial estimate of the high-resolution image $f^{(0)}$, by interpolating the first observed low-resolution image in the set by a scale factor S. Next $\vec{f_n}$ is computed by modifying the indexes of f^{n-1} such that the translation parameters will fulfill dx $\epsilon[0,1/Q,...,(Q-1)/Q]$ and dy

$\epsilon[0, 1/Q, ..., (Q-1)/Q]$. The translation parameters are then quantized. Starting with $k = 1$, the set of low-resolution input images are simulated by the following equation:

$$g_k^{(n)}(u, v) = \frac{(f_n(uS, vS) * \vec{h_{i,j}^{PP\_PSF}}(x, y) \downarrow S}{\sum_{s,t} h_{i,j}^{PP\_PSF}(s, t)}. \tag{7.7}$$

The differences between each simulated image and its corresponding observed image are now used to update the initial guess image $f^{(0)}$. This update scheme is modelled by the following equation:

$$f^{(n+1)}(x, y) = \vec{f}_n + C \cdot (Dg_k(u, v) \uparrow S) * h_{k-1-i,k-1-j}^{PP\_BP}(x, y), \tag{7.8}$$

where C is a convergence constant factor. The above process is then repeated by increasing the value of k.

7.3 Zomet, Rav-Acha, and Peleg's Approach

The problem associated with many super-resolution algorithms is their sensitivity to the presence of outliers in different regions of an image. These outliers may include noise, motion blur, moving objects and motion errors [22]. The work by Zomet, Rav-Acha, and Peleg [22] attempts to improve spatial resolution in these regions of an image where outliers are present. In order to accomplish this, a robust median estimator is combined in the resolution enhancement procedure. This resolution enhancement procedure follows the same process as Irani and Peleg [8], by back-projecting differences between a simulated low-resolution image and an observed image onto the initial guess of the high-resolution image.

Given n observed low-resolution images $g_1, ..., g_n$, Zomet, Rav-Acha, and Peleg simulate the low-resolution images by the following equation:

$$\vec{Y_k} = D_k C_k F_k \vec{X} + \vec{E_k}, \tag{7.9}$$

where

- $\vec{Y_k}$ is the k^{th} simulated low-resolution image,
- $\vec{X}$ is the high-resolution image that the algorithm is trying to find,
- $\vec{E_k}$ is an additive noise term that is normally distributed,
- F_k is the 2D transformation that maps $\vec{X}$ to $\vec{Y_k}$,

- C_k is a blurring operator, and
- D_k is a down-sampling operator.

Now the total squared error associated with simulating the above process for each low-resolution image, by resampling the high resolution image $\vec{X}$, is given by

$$L(\vec{X}) = \frac{1}{2} \sum_{k=1}^{n} \| \vec{Y_k} - D_k C_k F_k \vec{X} \|_2^2 . \tag{7.10}$$

If we take the derivative of L with respect to $\vec{X}$, we find that the gradient of L is the sum of gradients computed over the simulated low-resolution images. This is illustrated by the following equations:

$$\vec{B_k} = F_k^T C_k^T D_k^T (D_k C_k F_k \vec{X} - \vec{Y_k}), \tag{7.11}$$

$$\nabla L(\vec{X}) = \sum_{k=1}^{n} \vec{B_k}, \tag{7.12}$$

where $\vec{B}$ is a difference image back-projected onto the initial guess of the high-resolution image.

Zomet, Rav-Acha, and Peleg use a modified version of the iterative back-projection update scheme developed by Irani and Peleg [8]. This modified version updates the initial estimate in each iteration by

$$\vec{X^{n+1}} = \vec{X^n} + \lambda \nabla L(\vec{X}), \tag{7.13}$$

where λ is a scale factor defining the step size in the direction of the gradient.

Finally, robustness in the presence of outliers is now introduced into the iterative update scheme, by replacing the sum of images in Equation 7.12, with a scaled pixel-wise median as follows:

$$\nabla L(\vec{X})(x, y) \approx n \cdot median \left\{ \vec{B_k}(x, y) \right\}_{k=1}^{n} . \tag{7.14}$$

CHAPTER 8

Implementation of a Super-Resolution Algorithm

Many current super-resolution algorithms are based on the work by Irani and Peleg [7, 8]. As a result, we have implemented a super-resolution algorithm developed by Irani and Peleg [8]. The details of this implementation are presented in this chapter.

8.1 Irani and Peleg's Approach

In our implementation of the super-resolution algorithm, we assume that each observed low-resolution image has already been registered with the reference image. As a result, the super-resolution algorithm requires the set of all observed low-resolution images, registered low-resolution images, and the computed homographies as inputs.

Firstly an initial high-resolution estimate image is generated [8]. In order to accomplish this, an average of all the registered low-resolution images is taken. The averaged image is then up-sampled according to a specified scale factor using bilinear interpolation, in order to generate the high-resolution estimate.

The next step is to create a set of simulated low-resolution images that correspond to the observed low-resolution images [8]. The homography that maps an observed image back into the same spatial correspondence as the reference image is applied to the initial estimate. The initial estimate is then convolved with a blurring operator.

The blurring operator chosen is a point spread function that is based around the Butterworth filter. In MATLAB, the blurring operator has been provided by the function *psf2()* [10]. The *psf2()* function requires the user to specify the filter size, angle of rotation of the filter, length and width of the filter, and the squareness of the filter shape.

The Butterworth filter acts as a low-pass filter, thus attenuating high-frequencies [11]. As high-frequencies are filtered out fine detail in the image is removed, in effect blurring the image. The Butterworth filter is a 'bump' of height one that is centred on the origin. The Butterworth filter can be expressed by the following equation:

$$F = \frac{1.0}{1.0 + (r/cutoff)^{2n}}, \tag{8.1}$$

where

- r is the distance from the origin,
- cutoff is the radius of the bump, and
- n is a positive integer which controls the sharpness of the boundary.

The blurring operator is applied to the image using the *filter2()* [10] function, which performs convolution in the spatial domain. Each simulated low-resolution image is finally generated by down-sampling this blurred, transformed version of the initial guess image, according to the specified scale factor. Note that bilinear interpolation is used to interpolate pixel values.

Each of these simulated low-resolution images are now compared to its corresponding observed low-resolution image by taking the difference image. This difference image is then up-sampled to the size of the initial estimate image. The inverse homography is applied to the difference image, bringing it as close as possible in spatial correspondence to the initial estimate image. The blurring operation performed earlier is now reversed by convolving this transformed difference image with a back-projection kernel. Clearly the back-projection kernel p, is determined by the blurring operator h.

The back-projection kernel chosen is the Wiener filter, which is used to restore the blurred image. In MATLAB, the Wiener filter is generated by the function *wienfilter()* [10]. This requires the user to specify the image to be processed, length and width of the point spread function, order of the Butterworth function used to construct the point spread function, angle of the major axis of the point spread function and the estimated noise to signal power ratio.

The Wiener filter is based on the model that a blurred image is the convolution of an ideal, sharp image with a point spread function of some kind [11]. This convolution can be expressed as follows

$$G = F * H,$$

where

- F represents the Fourier transform of the sharp image,
- G represents the Fourier transform of the blurred image, and
- H represents the Fourier transform of the point spread function.

In order to obtain an estimate of F, the Fourier transform of the sharp image, an inverse filter can be applied as follows

$$F = G/H,$$

where

- 1/H is the inverse filter.

However, several problems exist with this type of filter [11]. As a result, the Wiener filter contains a modifying term on the right hand side. In its simplest form, the Wiener filter can be expressed by the following equation:

$$W = \frac{1}{H} \cdot \frac{|H|^2}{(|H|^2 + K)}, \tag{8.2}$$

where

- H is the Fourier transform of the point spread function, and
- K is a constant representing the ratio of the noise power spectrum to the signal power spectrum.

From the above equation, the modifying term on the right hand side stops the overall filter value from becoming infinity when the magnitude of H is small or zero. Likewise, the modifying term also returns the overall filter value to 1/H, when the magnitude of H is large relative to K.

This process is repeated for each difference image and the total difference image is taken and then averaged by the number of images. Once averaged, the results are back-projected to the initial estimate which is then updated. This iterative update scheme continues until a specified number of iterations has been exceeded or an error threshold has been reached.

CHAPTER 9

Visual Quality of Reconstructed Images

In this chapter, we revisit our earlier hypothesis that accurate image registration is vital in the super-resolution process. We begin by discussing the experimental framework that will be used to evaluate this hypothesis. From there, the final results will be presented.

9.1 Methodology

9.1.1 Test Cases

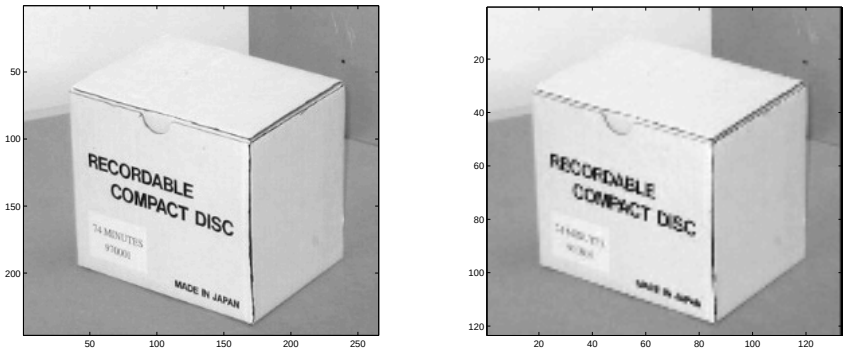
We revisit our earlier hypothesis that accurate image registration is vital in the super-resolution process. As a result, we hypothesize the following statements:

1. The visual quality of a reconstructed image is higher if perfect registration is achieved; and
2. The visual quality of a reconstructed image is higher if the RANSAC algorithm is employed in the registration process, and the transform that exists between images contains a higher degree of freedom. This is compared to employing a registration algorithm based on an optimization approach in the registration process.

The second hypothesis is a direct result from our earlier work on image registration. We have proved that the RANSAC algorithm achieves greater registration accuracy in the presence of higher degree transforms such as affine transforms. This is in comparison to registration algorithms based on an optimization approach. Therefore, if we were to reconstruct a high-resolution image in which a higher degree transform exists between the observed images, its visual quality improves if the RANSAC algorithm is employed in the registration process.

Ideally, a set of observed low-resolution images taken of the same scene are obtained by using a camera. However, in the following experiments these images

are generated by down-sampling a picture and text image by a factor of two, and then applying a transform according to three test cases. Figure 9.1 illustrates the down-sampling process for the picture image. We use the same text image from earlier experiments. Having obtained a set of observed low-resolution images, we now use the super-resolution algorithm to reconstruct a high-resolution image.



(a) Original image (b) Observed low-resolution version

Figure 9.1: Down-sampling process used to generate a set of observed low-resolution images.

The first test case involves translating the reference image by a combination of small pixel shifts in the x and y directions, to generate nine observed images. Given that dx and dy are the translational shifts in the x and y directions respectively, the following table represents the combination of small pixel shifts applied to the reference image:

Translations	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8	Image 9
dx	0	1	2	0	1	2	0	1	2
dy	0	0	0	1	1	1	2	2	2

Table 9.1: Combination of small pixel shifts used to create the observed images.

The second test case involves rotating the reference image by an unknown random amount between 0 and 40 degrees. Before the super-resolution process occurs, the RANSAC algorithm is used to register the images and to calculate the homographies.

The third test case involves applying an affine transform to the reference image. The scaling in x with respect to y, denoted by a, is randomly generated between 0.7 and 1.0. In a similar fashion, the cotangent of the shearing angle

θ, denoted by b, is randomly generated between 0 and 0.4. Before the super-resolution process occurs, both implemented registration algorithms are used to register the images and to calculate the homographies.

During the super-resolution process for each test case, the parameters for the Butterworth and Wiener filters are altered systematically. Only two parameters are altered since they directly influence the result of the super-resolution process. The order of the Butterworth and Wiener filters, denoted by n, is increased from two to four. The estimated noise to signal power ratio in the Wiener filter, denoted by K, is increased from 0.03 to 0.05 by an increment of 0.01. Ideally we would like K to be as small as possible. The super-resolution algorithm is set to terminate at five iterations.

9.2 Results

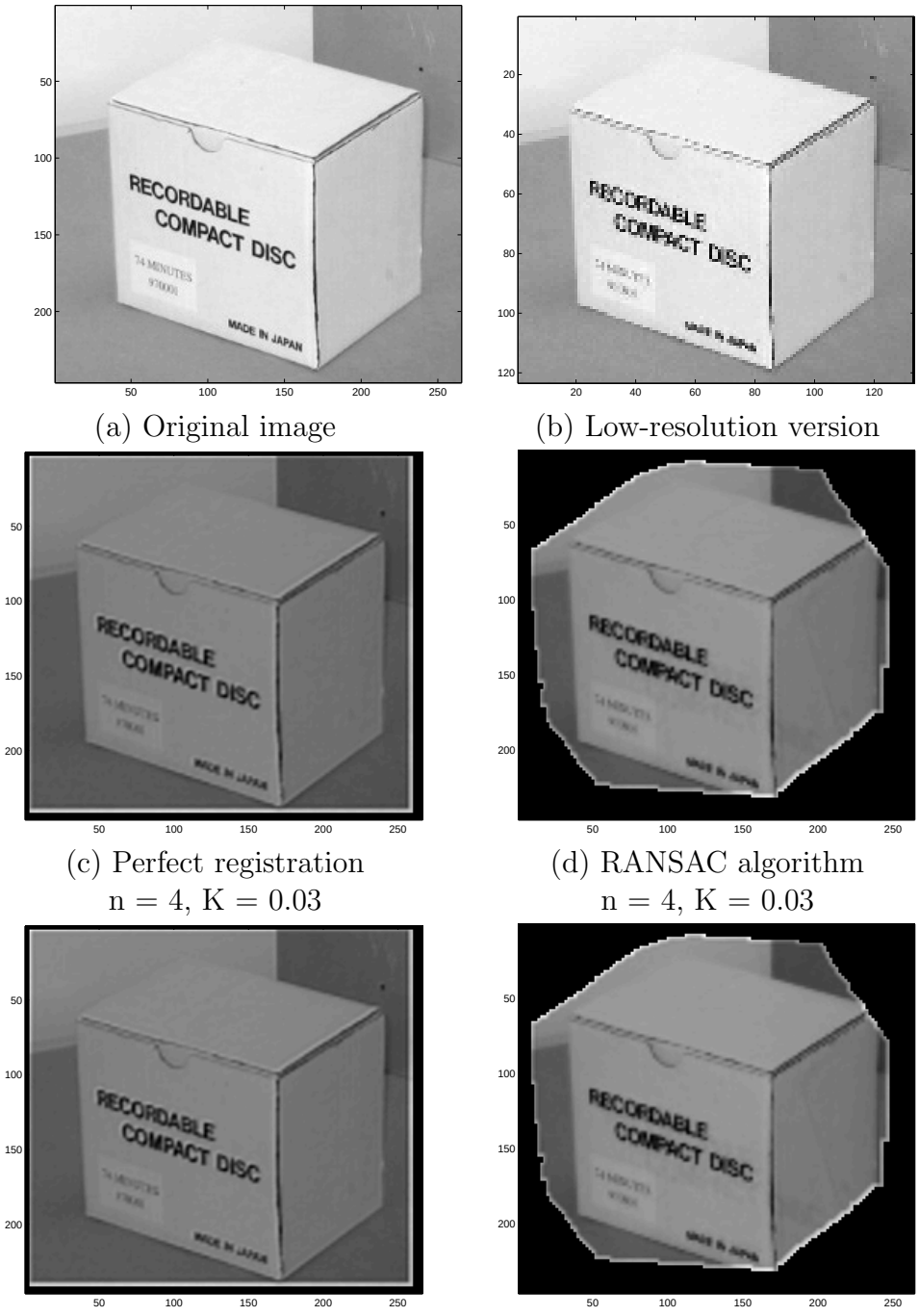
9.2.1 Reconstructing Picture Images

Visually comparing the results from the first and second test cases confirms our earlier hypothesis. That is, the visual quality of a reconstructed image is higher if perfect registration is achieved. The results from both test cases are shown side-by-side in Figure 9.2. Note that not all results have been shown. In these cases the error values obtained after each iteration in the super-resolution process, do not decrease but rather tend to infinity. Thus the visual quality of the reconstructed image becomes degraded after each iteration.

The first row of images shown in the figure include the original image, and the low-resolution version we are trying to reconstruct. Underneath this on the left hand side, are the reconstructed high-resolution images when the homographies are precisely known. That is, when perfect registration is achieved. On the right hand side are the reconstructed high-resolution images when the RANSAC algorithm is employed in the registration process. Full listings of results, including error values after each iteration, can be found in Appendix B.

If we inspect both columns of images carefully, the writing on the box is sharper and slightly clearer in the images where perfect registration is achieved. In addition, the error values obtained after each iteration in the super-resolution process are much higher for the RANSAC algorithm.

Similarly, the results from the third test case, as shown in Figure 9.3, confirm the second hypothesis. That is, the visual quality of a reconstructed image is higher if the RANSAC algorithm, as compared to an optimization approach, is employed in the registration process. This is especially true if the transform



(a) Original image

(b) Low-resolution version

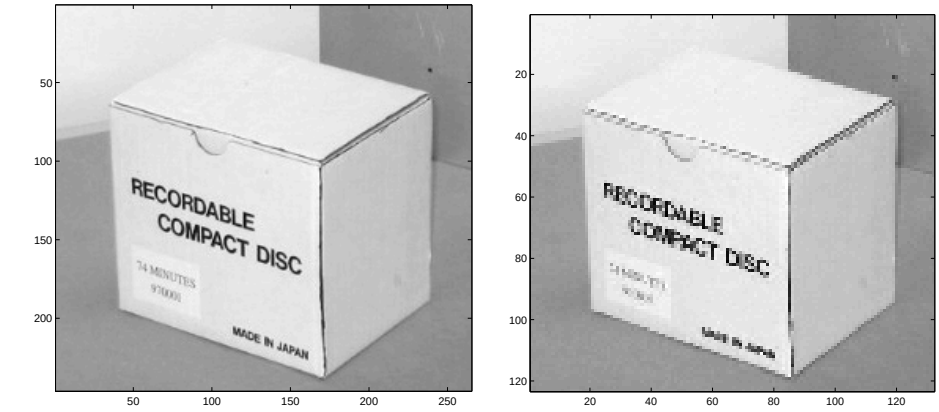
(c) Perfect registration
n = 4, K = 0.03

(d) RANSAC algorithm
n = 4, K = 0.03

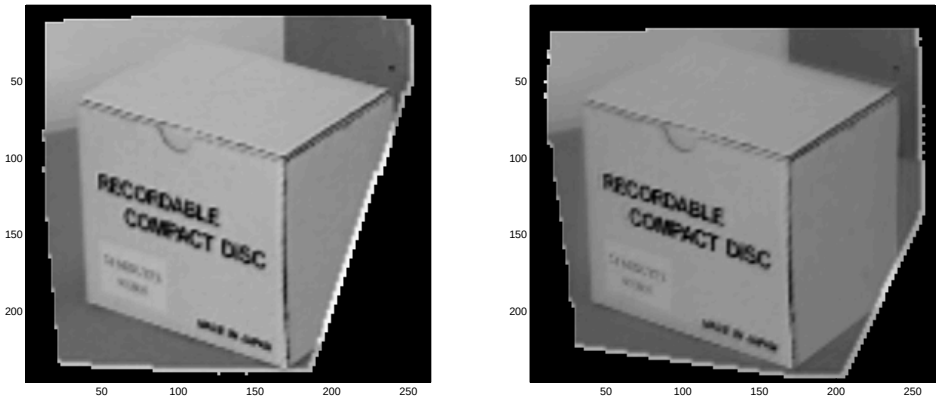
(e) Perfect registration
n = 4, K = 0.05

(f) RANSAC algorithm
n = 4, K = 0.05

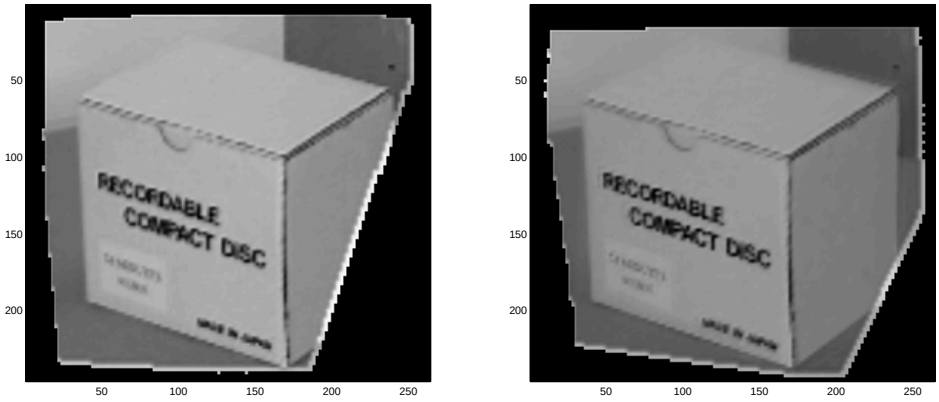
Figure 9.2: Visual quality of reconstructed picture images is higher when perfect registration is achieved.



(a) Original image (b) Low-resolution version



(c) RANSAC algorithm n = 4, K = 0.03 (d) Minimization of the SSD n = 4, K = 0.03



(e) RANSAC algorithm n = 4, K = 0.05 (f) Minimization of the SSD n = 4, K = 0.05

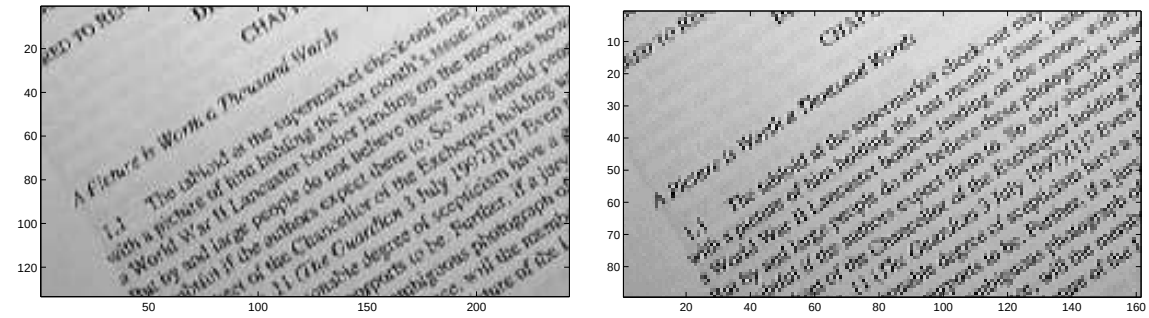
Figure 9.3: Visual quality of reconstructed picture images is higher when the RANSAC algorithm is employed in the registration process, for higher degree transforms.

that exists between images contains a higher degree of freedom. The error values for the RANSAC algorithm are visibly lower than the optimization approach, as shown in Appendix B.

9.2.2 Reconstructing Text Images

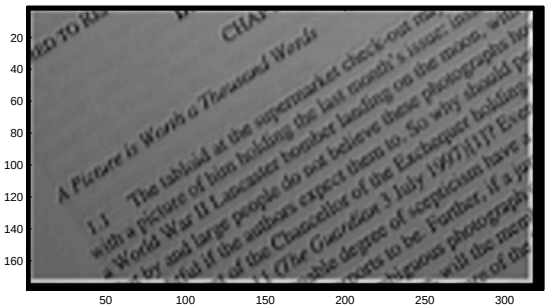
Similar results are obtained when reconstructing text images. We found that when perfect registration is achieved, the visual quality of the reconstructed text images is higher than if the RANSAC algorithm is employed. In addition, for higher degree transforms such as affine transforms, employing RANSAC in the registration process improves the visual quality of the high-resolution text images. This is compared to employing a registration algorithm based on an optimization approach in the registration process. These results are shown in Figure 9.4 and Figure 9.5. The error values listed in Appendix B also reinforce both hypotheses, since the values for perfect registration are much lower than the RANSAC algorithm. Likewise, the error values for the RANSAC algorithm are much lower than the algorithm based on an optimization approach.

Since the results confirm the truth of both hypotheses, we can highlight the importance accurate image registration is to the super-resolution process. That is, the visual quality of reconstructed images is dependent on the accuracy of the image registration process.

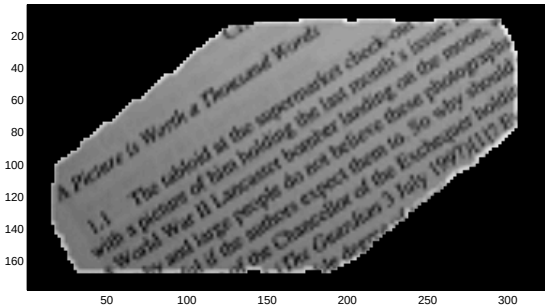


(a) Original image

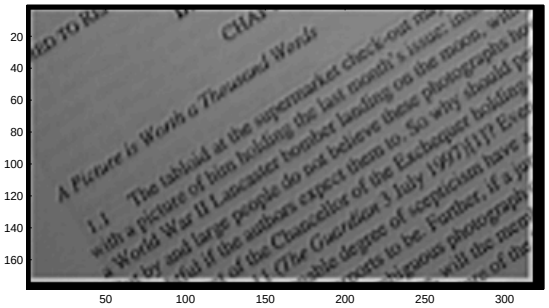
(b) Low-resolution version



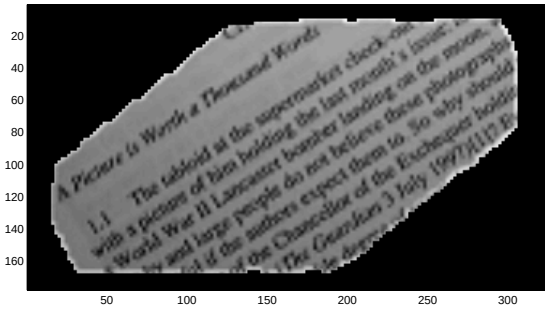
(c) Perfect registration
n = 4, K = 0.03



(d) RANSAC algorithm
n = 4, K = 0.03

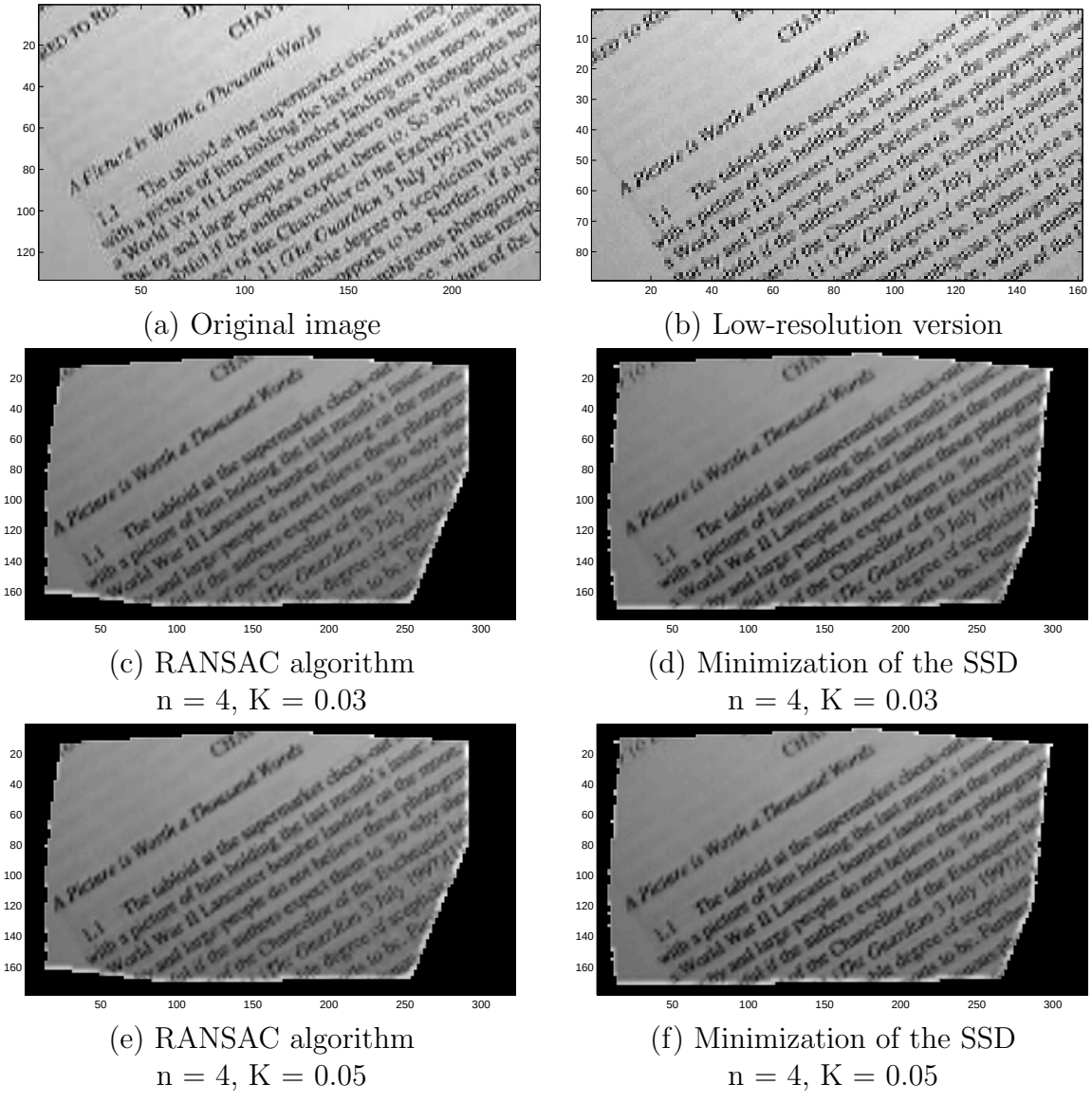


(e) Perfect registration
n = 4, K = 0.05



(f) RANSAC algorithm
n = 4, K = 0.05

Figure 9.4: Visual quality of reconstructed text images is higher when perfect registration is achieved.



(a) Original image

(b) Low-resolution version

(c) RANSAC algorithm
n = 4, K = 0.03

(d) Minimization of the SSD
n = 4, K = 0.03

(e) RANSAC algorithm
n = 4, K = 0.05

(f) Minimization of the SSD
n = 4, K = 0.05

Figure 9.5: Visual quality of reconstructed text images is higher when the RANSAC algorithm is employed in the registration process, for higher degree transforms.

CHAPTER 10

Conclusion

Our original hypothesis stated that accurate image registration is vital in the super-resolution process. That is, the visual quality of a reconstructed high-resolution image improves when accurate image registration is achieved. Having conducted numerous experiments, we found that the results obtained clearly prove and confirm this hypothesis.

We began by comparing the registration accuracy of two widely used registration algorithms. The first algorithm, based on an optimization approach, suffers in the presence of higher degree transforms such as affine transforms. This is attributed to the increased number of parameters requiring optimization. In contrast, the second algorithm based on the RANSAC algorithm, does not suffer in the presence of higher degree transforms. Instead, it achieves much greater registration accuracy. From there, we analyzed the registration accuracy of each algorithm when Gaussian noise is added to the input images. Both algorithms rely on the SSD to either measure the difference between two images or to obtain a set of putative correspondences. As a result, no appreciable difference is seen since the SSD is robust against uniform white Gaussian noise [9].

The study then moved to the super-resolution process. In this study, we obtained reconstruction results for an image when perfect registration was achieved. That is, the homographies were known before hand. Likewise, we obtained reconstruction results for an image when the RANSAC algorithm was employed in the registration process. Comparing the two sets of results, we found that the visual quality of the reconstructed image was higher when perfect registration was achieved. This is reinforced by the lower error values obtained after each iteration for perfect registration. We also found that the visual quality of reconstructed images was higher, when the RANSAC algorithm was employed in the registration process as compared to the optimization approach. Therefore, accurate image registration is vital in the super-resolution process.

Future extensions of this work can take many forms. This includes improving the registration accuracy to sub-pixel accuracy. In order to accomplish this,

different optimization routines may be employed, which overcome the limitation of higher degree transforms. The mechanism to determine the set of control points can also be improved since it is error prone and requires user intervention. A solution is to use automated methods [21]. Different objective criteria may also be used, such as the mutual information of the joint pixel distributions of two images [5].

In the super-resolution process, the visual quality of reconstructed high-resolution images can be improved. This includes incorporating a robust median estimator into the imaging model, used to back-project the image differences onto the initial high resolution estimate [22]. Likewise, Polyphase filters can be introduced into the simulation of the low-resolution images and when back-projecting image differences [4].

APPENDIX A

Results on Registering Synthetic Images

A.1 Small Translations and Rotations Applied to a Picture Image

A.1.1 Results for the Minimization of the SSD Algorithm

x Shift	y Shift	Θ Rotation (Radians)	Transformation Error (e-015)
1	1	0.2	5.2281
1	2	0.4	19.153
1	3	0.6	15.306
1	4	0.8	16.791
1	5	1.0	15.858
2	1	0.2	5.4872
2	2	0.4	11.931
2	3	0.6	18.435
2	4	0.8	16.936
2	5	1.0	31.337
3	1	0.2	7.9458
3	2	0.4	10.762
3	3	0.6	26.560
3	4	0.8	25.444
3	5	1.0	52.943
4	1	0.2	4.7248
4	2	0.4	7.3706
4	3	0.6	11.942
4	4	0.8	25.478
4	5	1.0	38.420
5	1	0.2	7.5521
5	2	0.4	8.7641
5	3	0.6	25.828
5	4	0.8	30.602
5	5	1.0	28.655

A.1.2 Results for the RANSAC Algorithm

x Shift	y Shift	Θ Rotation (Radians)	Number of Inliers	Number of Putative Correspondences	Transformation Error (e-015)
1	1	0.2	49	103	4.0506
1	2	0.4	23	72	10.006
1	3	0.6	49	97	15.738
1	4	0.8	51	115	21.987
1	5	1.0	53	111	19.733
2	1	0.2	28	95	6.0675
2	2	0.4	43	102	7.1633
2	3	0.6	42	94	16.706
2	4	0.8	41	92	30.573
2	5	1.0	38	107	36.626
3	1	0.2	46	107	4.6497
3	2	0.4	47	111	9.1111
3	3	0.6	49	102	16.532
3	4	0.8	43	108	23.318
3	5	1.0	46	91	20.120
4	1	0.2	38	92	5.3807
4	2	0.4	36	83	9.1011
4	3	0.6	28	80	17.936
4	4	0.8	41	99	27.743
4	5	1.0	56	110	63.537
5	1	0.2	47	103	7.5912
5	2	0.4	39	100	12.926
5	3	0.6	41	90	22.796
5	4	0.8	41	83	19.256
5	5	1.0	28	70	35.074

A.2 Affine Transform Applied to a Picture Image

A.2.1 Results for the Minimization of the SSD Algorithm

a	b	Transformation Error (e-015)
0.7	0.1	2.9123
0.7	0.2	2.7948
0.7	0.3	1.6296
0.7	0.4	0.91749
0.8	0.1	1.9673
0.8	0.2	7.0883
0.8	0.3	0.28407
0.8	0.4	0.33046
0.9	0.1	5.9400
0.9	0.2	3.4074
0.9	0.3	0.60399
0.9	0.4	3.7337
1.0	0.1	2.1224
1.0	0.2	7.1414
1.0	0.3	1.4395
1.0	0.4	3.5622

A.2.2 Results for the RANSAC Algorithm

a	b	Number of Inliers	Number of Putative Correspondences	Transformation Error (e-015)
0.7	0.1	38	102	0.94268
0.7	0.2	38	86	0.091513
0.7	0.3	42	89	0.33512
0.7	0.4	40	102	0.10642
0.8	0.1	41	97	0.28125
0.8	0.2	38	76	0.19541
0.8	0.3	53	113	0.49596
0.8	0.4	52	108	0.42881
0.9	0.1	52	119	0.13034
0.9	0.2	42	117	0.58776
0.9	0.3	50	109	0.30032
0.9	0.4	40	102	0.57868
1.0	0.1	56	111	0.13555
1.0	0.2	75	140	0.38568
1.0	0.3	41	107	1.2127
1.0	0.4	47	93	0.19435

A.3 Small Translations and Rotations Applied to a Text Image

A.3.1 Results for the Minimization of the SSD Algorithm

x Shift	y Shift	Θ Rotation (Radians)	Transformation Error (e-015)
1	1	0.2	4.3299
1	2	0.4	7.6471
1	3	0.6	22.309
1	4	0.8	18.717
1	5	1.0	17.167
2	1	0.2	3.2844
2	2	0.4	7.3626
2	3	0.6	13.724
2	4	0.8	16.681
2	5	1.0	19.598
3	1	0.2	2.9658
3	2	0.4	9.8340
3	3	0.6	10.296
3	4	0.8	16.182
3	5	1.0	11.721
4	1	0.2	2.8444
4	2	0.4	7.6389
4	3	0.6	12.961
4	4	0.8	10.689
4	5	1.0	35.225
5	1	0.2	2.8206
5	2	0.4	7.8167
5	3	0.6	8.0082
5	4	0.8	39.230
5	5	1.0	18.267

A.3.2 Results for the RANSAC Algorithm

x Shift	y Shift	Θ Rotation (Radians)	Number of Inliers	Number of Putative Correspondences	Transformation Error (e-015)
1	1	0.2	157	166	4.0582
1	2	0.4	101	103	13.204
1	3	0.6	182	182	7.2618
1	4	0.8	123	154	12.040
1	5	1.0	95	99	18.591
2	1	0.2	183	194	5.3608
2	2	0.4	143	143	11.460
2	3	0.6	214	220	2.0376
2	4	0.8	91	93	28.662
2	5	1.0	141	162	39.242
3	1	0.2	159	186	5.4377
3	2	0.4	150	168	9.9501
3	3	0.6	150	158	15.154
3	4	0.8	176	184	20.043
3	5	1.0	120	136	11.693
4	1	0.2	187	228	4.6473
4	2	0.4	196	197	7.4152
4	3	0.6	162	176	11.573
4	4	0.8	154	165	13.818
4	5	1.0	167	170	26.021
5	1	0.2	205	207	5.2863
5	2	0.4	136	145	5.3909
5	3	0.6	188	188	9.2182
5	4	0.8	136	142	18.922
5	5	1.0	148	177	30.003

A.4 Affine Transform Applied to a Text Image

A.4.1 Results for the Minimization of the SSD Algorithm

a	b	Transformation Error (e-015)
0.7	0.1	1.6507
0.7	0.2	1.4477
0.7	0.3	0.30364
0.7	0.4	0.29950
0.8	0.1	1.2450
0.8	0.2	0.25555
0.8	0.3	1.3942
0.8	0.4	0.12904
0.9	0.1	0.54583
0.9	0.2	0.92881
0.9	0.3	2.8481
0.9	0.4	0.88882
1.0	0.1	1.4407
1.0	0.2	1.0387
1.0	0.3	0.17318
1.0	0.4	0.83355

A.4.2 Results for the RANSAC Algorithm

a	b	Number of Inliers	Number of Putative Correspondences	Transformation Error (e-015)
0.7	0.1	87	97	0.12923
0.7	0.2	89	93	0.13774
0.7	0.3	116	123	0.21430
0.7	0.4	111	113	0.30920
0.8	0.1	205	211	0.087909
0.8	0.2	191	194	0.15449
0.8	0.3	153	156	0.11385
0.8	0.4	164	168	0.17142
0.9	0.1	206	210	0.074697
0.9	0.2	168	190	0.10655
0.9	0.3	159	197	0.15371
0.9	0.4	362	373	0.082684
1.0	0.1	255	265	0.15577
1.0	0.2	121	123	0.27256
1.0	0.3	210	218	0.12615
1.0	0.4	120	124	0.26230

A.5 Gaussian Noise Added to a Translated and Rotated Picture Image

A.5.1 Results for the Minimization of the SSD Algorithm

x Shift	y Shift	Θ Rotation (Radians)	Transformation Error (e-015)
1	1	0.2	6.5236
1	2	0.4	22.244
1	3	0.6	11.827
1	4	0.8	28.278
1	5	1.0	17.235
2	1	0.2	4.0540
2	2	0.4	10.062
2	3	0.6	17.225
2	4	0.8	30.512
2	5	1.0	25.909
3	1	0.2	5.2697
3	2	0.4	14.221
3	3	0.6	12.726
3	4	0.8	10.540
3	5	1.0	42.010
4	1	0.2	13.178
4	2	0.4	8.6339
4	3	0.6	24.221
4	4	0.8	16.417
4	5	1.0	44.503
5	1	0.2	3.7275
5	2	0.4	13.940
5	3	0.6	16.537
5	4	0.8	15.023
5	5	1.0	33.321

A.5.2 Results for the RANSAC Algorithm

x Shift	y Shift	Θ Rotation (Radians)	Number of Inliers	Number of Putative Correspondences	Transformation Error (e-015)
1	1	0.2	17	60	4.5616
1	2	0.4	13	77	6.9763
1	3	0.6	11	64	12.405
1	4	0.8	10	65	18.368
1	5	1.0	10	58	19.402
2	1	0.2	9	65	6.6810
2	2	0.4	13	78	13.614
2	3	0.6	11	64	11.999
2	4	0.8	11	65	20.783
2	5	1.0	12	64	56.152
3	1	0.2	10	58	4.6733
3	2	0.4	12	60	10.194
3	3	0.6	10	54	14.013
3	4	0.8	11	81	34.506
3	5	1.0	9	64	27.531
4	1	0.2	17	65	8.1905
4	2	0.4	10	73	8.1589
4	3	0.6	8	61	10.558
4	4	0.8	10	65	11.271
4	5	1.0	10	68	30.097
5	1	0.2	12	67	3.0057
5	2	0.4	10	75	12.461
5	3	0.6	11	69	19.216
5	4	0.8	9	65	25.705
5	5	1.0	12	71	21.762

A.6 Gaussian Noise Added to an Affine Transformed Picture Image

A.6.1 Results for the Minimization of the SSD Algorithm

a	b	Transformation Error (e-015)
0.7	0.1	2.7905
0.7	0.2	2.9916
0.7	0.3	4.6894
0.7	0.4	2.6391
0.8	0.1	1.5863
0.8	0.2	0.69182
0.8	0.3	4.9619
0.8	0.4	4.0140
0.9	0.1	5.3180
0.9	0.2	3.2627
0.9	0.3	2.9468
0.9	0.4	2.8636
1.0	0.1	1.3982
1.0	0.2	2.3321
1.0	0.3	1.1588
1.0	0.4	4.9482

A.6.2 Results for the RANSAC Algorithm

a	b	Number of Inliers	Number of Putative Correspondences	Transformation Error (e-015)
0.7	0.1	12	97	1.3390
0.7	0.2	13	97	5.0334
0.7	0.3	9	57	1.7229
0.7	0.4	12	72	1.3830
0.8	0.1	8	46	0.97341
0.8	0.2	10	53	1.7349
0.8	0.3	12	78	2.7834
0.8	0.4	7	54	6.2389
0.9	0.1	10	68	1.7517
0.9	0.2	9	66	5.3690
0.9	0.3	11	76	0.39500
0.9	0.4	9	53	4.1311
1.0	0.1	9	65	4.2277
1.0	0.2	12	66	0.70263
1.0	0.3	10	47	1.5561
1.0	0.4	10	72	0.62604

APPENDIX B

Results on Reconstructing Synthetic Images

B.1 Perfect Registration for a Translated Picture Image

		Error Value				
n	K	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
2	0.03	428.5856	455.5329	475.0609	489.3373	499.8156
2	0.04	428.5856	453.3441	471.4239	484.7315	494.5595
2	0.05	428.5856	451.3563	468.1010	480.5008	489.7090
4	0.03	428.6199	423.2881	419.0317	415.6339	412.9192
4	0.04	428.6199	423.3355	419.1061	415.7214	413.0102
4	0.05	428.6199	423.3821	419.1795	415.8078	413.1005

B.2 Registration by RANSAC for a Rotated Picture Image

		Error Value				
n	K	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
2	0.03	761.5357	772.8726	782.7352	791.3324	798.8390
2	0.04	761.5357	772.1932	781.4971	789.6335	796.7593
2	0.05	761.5357	771.5728	780.3623	788.0715	794.8417
4	0.03	760.6600	760.1847	759.7352	759.3095	758.9057
4	0.04	760.6600	760.1872	759.7399	759.3161	758.9141
4	0.05	760.6600	760.1897	759.7446	759.3228	758.9225

B.3 Registration by RANSAC for an Affine Transformed Picture Image

		Error Value				
n	K	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
2	0.03	478.3825	489.8693	499.8823	508.6329	516.2953
2	0.04	478.3825	489.2301	498.7109	507.0167	514.3067
2	0.05	478.3825	488.6364	497.6209	505.5103	512.4502
4	0.03	477.6380	476.8471	476.1155	475.4374	474.8074
4	0.04	477.6380	476.8539	476.1281	475.4549	474.8290
4	0.05	477.6380	476.8607	476.1405	475.4721	474.8502

B.4 Registration by Minimization of the SSD for an Affine Transformed Picture Image

		Error Value				
n	K	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
2	0.03	547.4379	556.2669	564.0258	570.8543	576.8709
2	0.04	547.4379	555.7868	563.1402	569.6256	575.3517
2	0.05	547.4379	555.3401	562.3148	568.4788	573.9319
4	0.03	546.8336	546.3700	545.9396	545.5390	545.1655
4	0.04	546.8336	546.3740	545.9469	545.5491	545.1780
4	0.05	546.8336	546.3778	545.9540	545.5591	545.1903

B.5 Perfect Registration for a Translated Text Image

		Error Value				
n	K	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
2	0.03	381.1270	403.9488	420.5387	432.6920	441.6239
2	0.04	381.1270	402.1073	417.4713	428.8009	437.1786
2	0.05	381.1270	400.4335	414.6672	425.2258	433.0755
4	0.03	381.1569	376.5595	372.8964	369.9779	367.6507
4	0.04	381.1569	376.6002	372.9602	370.0525	367.7282
4	0.05	381.1569	376.6402	373.0230	370.1263	367.8049

B.6 Registration by RANSAC for a Rotated Text Image

		Error Value				
n	K	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
2	0.03	711.9356	719.3484	725.8181	731.4729	736.4213
2	0.04	711.9356	718.9049	725.0081	730.3596	735.0568
2	0.05	711.9356	718.5002	724.2664	729.3368	733.7994
4	0.03	711.4002	711.1183	710.8521	710.6006	710.3626
4	0.04	711.4002	711.1195	710.8544	710.6038	710.3666
4	0.05	711.4002	711.1207	710.8567	710.6070	710.3706

B.7 Registration by RANSAC for an Affine Transformed Text Image

		Error Value				
n	K	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
2	0.03	383.0882	393.3794	402.3718	410.2524	417.1748
2	0.04	383.0882	392.8223	401.3473	408.8340	415.4235
2	0.05	383.0882	392.3031	400.3912	407.5090	413.7857
4	0.03	382.4869	381.8107	381.1902	380.6198	380.0946
4	0.04	382.4869	381.8165	381.2009	380.6345	380.1125
4	0.05	382.4869	381.8223	381.2113	380.6489	380.1301

B.8 Registration by Minimization of the SSD for an Affine Transformed Text Image

		Error Value				
n	K	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
2	0.03	501.6807	510.8968	518.9611	526.0289	532.2309
2	0.04	501.6807	510.3934	518.0348	524.7466	530.6488
2	0.05	501.6807	509.9247	517.1710	523.5492	529.1693
4	0.03	501.3811	500.7042	500.0719	499.4799	498.9243
4	0.04	501.3811	500.7101	500.0829	499.4953	498.9435
4	0.05	501.3811	500.7159	500.0937	499.5105	498.9625

APPENDIX C

Original Honours Proposal

Title: Image Registration Algorithms for Creating a Super Resolution Image from Multiple Low-Resolution Images

Author: Douglas Lim

Supervisor: Dr. Peter Kovesi

C.1 Background

When multiple images obtained from different sensors, different viewpoints, or at different times are taken of the same scene, they become distorted with respect to one another [2]. Image registration is the process of bringing these distorted images back into spatial alignment with one another. Registration involves finding the optimal transformation that maps each pixel in one image (input image) into the same spatial correspondence as the pixels in the second image (reference image).

Image registration plays a pivotal role in extracting crucial information in many different application areas. For example, in medical image analysis, breast tumours can be detected by subtracting two aligned images of the same breast taken some time apart [16]. In forensic imaging, image registration is the basis step in creating a higher-resolution image from a set of multiple low-resolution images. This process is known as super-resolution. Super-resolution allows forensic scientists to extract crucial information from a crime scene image that is otherwise barely visible to the human eye [18].

Most papers on super resolution assume images have already been perfectly registered. It is this registration step that is crucial for the success of these algorithms. Currently, there is a large body of research into optimizing and developing image registration algorithms. These include algorithms documented

by Brown [2]; Irani and Peleg [7]; Thevenaz, Ruttimann, and Unser [19]; and Gordon [16].

C.2 Aim

Using a software engineering approach, I aim to research numerous image registration algorithms that currently exist. Following this, I will implement some of these algorithms in MATLAB and then continuously refine them to improve registration quality. This refinement process will involve conducting numerous experiments.

I will build upon the work already done by Edward Suhendra in 2002 [18]. Edward implemented an image registration algorithm based on an optimization approach. The initial solution for the optimization was obtained using control points, manually entered by the user to find the transformation between images. I will implement a similar algorithm that will focus on investigating and improving the problems encountered with his implementation.

In addition, I aim to implement an image registration algorithm based on the RANdom SAmple Consensus (RANSAC) algorithm of Fischler and Bolles [1], and an image registration algorithm based on the mutual information between two images [5]. The image registration code developed will also be used as the basis for creating a higher resolution image from multiple low-resolution images taken of the same scene, based upon the super resolution technique proposed by Irani and Peleg [7]. In particular, I aim to refine the super-resolution algorithm implemented by Edward.

C.3 Method

The first step is to spend time researching literature on the current methods of image registration. Understanding the scope of this research will in turn help me to analyze the various image registration techniques. The next step will be to implement some of these algorithms using MATLAB.

Once the coding has been completed, numerous experiments will be conducted using synthetic data and images produced by Adobe Photoshop. Both picture and text images will be used. These experiments will help to refine the code to allow for small or large displacements between images and the degree of transformation between images.

Weeks	Uni	Event
10-13	1-4	Research image registration algorithms
14-16	5-7	Implement a registration algorithm based on manual control points
17	Break	Conduct experiments
18-20	8-10	Conduct experiments
21-22	11-12	Improve code
23	13	Implement a registration algorithm based on the RANSAC algorithm
24		Exam study
25-26		Exams
27	Break	Conduct experiments and refine code
28-29	Break	Implement a registration algorithm based on mutual information
30-31	1-2	Conduct experiments and refine code
32	3	Refine Edward's super resolution code
33-34	4-5	Conduct experiments and start dissertation
35-38	6-9	Dissertation
39	Break	Draft dissertation due
40	Break	Dissertation
41-42	10-11	Dissertation and project seminar preparation
43-44	12-13	Project seminar to be given
45	Break	Final dissertation due

C.4 Software and Hardware Requirements

Software required:
MATLAB - MATLAB will be used to implement the image registration and super resolution algorithms, including the processing of images.
Adobe Photoshop - This will be used as an image editing tool for producing different sample images.

Hardware required:
A computer with a 433 MHz Pentium processor or higher will be used to run MATLAB and to perform the image processing tasks. The computers in Lab G9 will be adequate for these purposes. However a computer with a faster processor is preferable to reduce compilation time.

Bibliography

[1] Bolles, R. C., and Fischler, M. A. A RANSAC-based approach to model fitting and its application to finding cylinders in range data. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI 1981* (1981), pp. 637–643.

[2] Brown, L. A survey of image registration techniques. *ACM Computing Surveys 24*, 4 (1992), 325–376.

[3] Capel, D., and Zisserman, A. Super-resolution enhancement of text image sequences. In *Proceedings of the 15th International Conference on Pattern Recognition* (2000), vol. 1, pp. 600–605.

[4] Cohen, B., Avrin, V., and Dinstein, I. Polyphase back-projection filtering for resolution enhancement of image sequences. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2000* (2000), vol. 4, pp. 2171–2174.

[5] Gilles, S. Description and experimentation of image matching using mutual information. Tech. rep., Department of Engineering Science. Oxford University, 1996.

[6] Hartley, R., and Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[7] Irani, M., and Peleg, S. Improving resolution by image registration. *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing 53* (May 1991), 231–239.

[8] Irani, M., and Peleg, S. Motion analysis for image enhancement: Resolution, occlusion, and transparency. In *Journal of Visual Communication and Image Representation* (December 1993), vol. 4, pp. 324–335.

[9] Kent, P. Multiresolution image registration. *IEEE Colloquium on Multiresolution Modelling and Analysis in Image Processing and Computer Vision 4* (April 1995), 1–6.

[10] Kovesi, P. Peter's MATLAB functions. *Available at http://www.cs.uwa.edu.au/∼pk/Research/MatlabFns/index.html* (October 2003).

[11] Kovesi, P. UWA computer science - computer vision 233.412. *Available at http://undergraduate.csse.uwa.edu.au/units/233.412/* (October 2003).

[12] MathWorks. Mathematics: Polynomials and interpolation: Interpolation function summary. *MATLAB Help Documentation* (June 2002).

[13] Ng, E., Larchev, G., and Williams, N. Automated extraction of interesting matching points. *Available at http://ise0.stanford.edu/class/ee368a_proj01/dropbox/project17/harris.html%/* (April 2003).

[14] Pagliari, G. M., and Greene, J. R. Image registration, parameter tuning and approximate function evaluation, using the genetic algorithm and digital image warping. In *Proceedings of the 4th IEEE AFRICON Conference* (September 1996), vol. 2, pp. 536–541.

[15] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. *Numerical Recipes — The Art of Scientific Computing*. Cambridge University Press, 1988.

[16] Sivaramakrishna, R., and Gordon, R. Image registration using minimization. *IEEE Proceedings of Communications, Power, and Computing 1* (1995), 181–184.

[17] Stone, H. S., Orchard, M. T., Chang, E.-C., and Martucci, S. A. A fast direct fourier-based algorithm for subpixel registration of images. *IEEE Transactions on Geoscience and Remote Sensing 39*, 10 (October 2001), 2235–2243.

[18] Suhendra, E. Super-resolution from multiple low resolution images. Master's thesis, School of Computer Science and Software Engineering, The University of Western Australia, 2002.

[19] Thevenaz, P., Ruttimann, U. E., and Unser, M. A pyramid approach to subpixel registration based on intensity. *IEEE Transactions on Image Processing 7* (January 1998), 27–41.

[20] Wang, T., Fadda, M., Nanetti, M., Marcacci, M., Martelli, S., and Visani, A. Interactive identification of marker position on 3D CT-based model. In *Proceedings of the 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Engineering Advances: New Opportunities for Biomedical Engineers* (November 1994), vol. 1, pp. 530–531.

[21] Zheng, Q., and Chellappa, R. A computational vision approach to image registration. In *Proceedings of the 11th IAPR International Conference on Pattern Recognition, Conference A: Computer Vision and Applications* (August 1992), vol. 1, pp. 193–197.

[22] Zomet, A., Rav-Acha, A., and Peleg, S. Robust super-resolution. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001* (2001), vol. 1, pp. 645–650.